

VTAP

Application Note - VTAP PRO BLE GATT server

VTAP Firmware from v2.2.6.0

VCF Firmware from v1.0.3.3

VTAP100-PRO-BW

Revised November 2024 v1.01

If you need help to set up or use your VTAP reader, beyond what is contained in this Application Note, then please contact our support team.

Email: vtap-support@dotorigin.com

Download the latest documentation and firmware from <https://vtapnfc.com>

Telephone UK and Europe: +44 (0) 1428 685861

Telephone North America and Latin America: +1 (562) 262-9642

If you have any feedback on setting up or using your VTAP reader or this documentation, then please contact our support team. The product is constantly being reviewed and improved and we value feedback about your experience.

Copyright 2024 Dot Origin Ltd. All rights reserved.

No part of this Application Note may be published or reproduced without the written permission of Dot Origin Ltd except for personal use. This Application Note relates to correct use of the VTAP reader only. No liability can be accepted under any circumstances relating to the operation of the user's own PC, network or infrastructure.

Dot Origin Ltd

Unit 7, Coopers Place Business Park, Combe Lane, Wormley

Godalming GU8 5SZ United Kingdom

+44 (0) 1428 685861

Contents

1 VTAP PRO BLE GATT server	1
1.1 Configuration of BLE GATT	1
1.2 VTAP BLE GATT custom service	2
2 Web Bluetooth API Example	5
3 iOS Implementation of BLE GATT	6
4 About Application Notes	7

1 VTAP PRO BLE GATT server

This feature provides an easy way for apps running on BLE capable devices, for example under iOS, Android or Windows, to connect to a VTAPI00-PRO-BW reader, be notified when data is available and to read the entire tap payload with a single read.

BLE GATT (Bluetooth Low Energy Generic ATtribute Profile) is how most Bluetooth Low Energy devices transfer data, by exposing *Services* which in turn define a number of *Characteristics* or data values. This is how standard BLE services such as HID keyboard are implemented, in this case defining a standard set of characteristics then adopted by the Bluetooth SIG standards organization. It is also common for devices to expose custom BLE services for specific purposes, where the standard services and their characteristics are not ideal or are not supported natively by host devices.

VTAP PRO BW readers can act as a BLE GATT server, from v1.0.3.3 release of VTAP connectivity firmware (VCF). This exposes a proprietary service that allows a host BLE device to read tap data payloads and to send tap responses to the VTAP in the form of action data. This feature is currently only available for users to configure on VTAP PRO readers in Local mode. Please contact vtap-support@dotorigin.com if you would like to discuss implementation of this feature on VTAP PRO readers in Cloud mode.

The advantage of using a custom BLE GATT service for this purpose, especially with an iPhone or iPad, is that the VTAPI00-PRO-BW reader does not need to be paired with the host device within its operating system, but can be directly discovered and connected from within the app itself using the iOS CoreBluetooth API. This behaviour is relatively simple to implement, is supported by Apple and requires no special app developer permissions or MFi certification to publish an app in Apple's App Store using this feature. The app can scan for and connect to the VTAP PRO reader directly and can read tap data without requiring keyboard focus. The app can also write back action responses, such as triggering user feedback (LEDs or buzzer) or controlling VTAP reader connected relays.

This option is an alternative to using the existing VTAP PRO BLE HID keyboard emulation feature to send tap data to a Bluetooth host device. The VTAP PRO BW cannot act as a both a BLE HID keyboard and a BLE VTAP GATT server at the same time.

1.1 Configuration of BLE GATT

There are new settings to enable and control the VTAP BLE GATT server feature which are used in the reader's `config.txt` file. For example:

Example: Changes to `config.txt` to enable a BLE GATT server

```
BTServerMode=1           ; BLE GATT server enabled
BTServerPIN=123456       ; Optional PIN set
BTServerName=MyVTAPName ; Optional change of advertised name from serial number
```

The `BTServerMode` defaults to `=0` which disables the custom GATT server, which means the `BTKeyboardMode` setting can then be used to enable the BLE keyboard. If `BTServerMode` is enabled, the BLE keyboard will not be enabled, even when `BTKeyboardMode=1` is set.

With `BTServerMode=1`, the custom service is enabled, then:

- `BTServerPIN` can be used to set a 6 digit PIN required to access the VTAP custom service. If this option is not present or set to `=0`, then no PIN is required to pair.
- `BTServerName` can be used to set your own Bluetooth advertised device name, overriding the default which is to advertise the VTAP reader's serial number as its device name.

1.2 VTAP BLE GATT custom service

The VTAP BLE GATT service has a 16-bit service UUID of `0xD0F0`. The client app can scan for BLE devices that advertise this service and potentially offer the user a list of devices which they can connect to.

The service provides 3 characteristics each with 16-bit UUIDs as follows:

- `0xD0F1` - Used to send data to the VTAP reader
- `0xD0F2` - Used to receive data from the VTAP reader
- `0xD0F3` - Status information

The data transferred on all three characteristics takes the form of a JSON string, which does not need to be zero terminated.

The data received (`0xD0F2`) and status (`0xD0F3`) characteristics will notify when they change. So when a tap is sent, the `0xD0F2` characteristic will be updated with the new tap data and the application will receive a notification of the change. It can read the tap data from the characteristic. The tap data will remain available until the connection is closed or a new tap is received.

The application can send a JSON string to the VTAP at any time by writing it to the `0xD0F1` characteristic. Normally this would be in response to a tap, to send a JSON string containing actions to perform, but this is not enforced. In the future, it may be used to support other commands or functions.

The `0xD0F3` characteristic can be read at any time, to query status information. The client app can also receive notifications when this value changes. The client application should use the notifications, after doing an initial read, rather than polling for status.

The tap data received (stored in the `0xD0F2` characteristic) takes this format:

Example: Tap data received in JSON (stored in the 0xD0F2 characteristic)

```
{
  "serialno": "CC123456",
  "payload": "3|DA2VbMkO_n-x61wN|32128290592065|Valued",
  "passtype": "G16",
  "passuid": "08E9460F"
}
```

This list of JSON attributes may be expanded in the future.

The client can send an action JSON message to provide user feedback, triggering LEDs and/or buzzer on the VTAP reader, by writing to the 0xD0F1 characteristic. This is in the same form as a response in the VTAP Cloud application:

Example: Action JSON message (written to the 0xD0F1 characteristic)

```
{
  "action": {
    "led": "?led ff0000,3000",
    "beep": "?beep 100,100,2"
  }
}
```

The “controls” and “inputs” arrays in a standard VTAP action JSON response are also supported within the action JSON, as well as the single “control” and “input” attributes defined in a VTAP Cloud application response. These attributes allow relays connected to VTAP PRO reader to be operated and allow messages to be send to other VTAP PRO interfaces.

The status on the 0xD0F3 characteristic is another JSON string:

Example: Status JSON string (written to the 0xD0F3 characteristic)

```
{
  "SerialNumber": "CC123456",
  "MaxData": 514,
  "AuthRequired": false
}
```

- The “SerialNumber” attribute provides the assigned serial number of the VTAP reader.
- The “MaxData” attribute lets the client know how much data can be sent in one go on the 0xD0F1 characteristic. Most BT stacks will negotiate this on connection, up to the maximum of 514 characters, but this can be restricted on some systems. If a blob of JSON longer than MaxData needs to be sent, the client can write the first part of the JSON blob up to MaxData characters, wait for a response acknowledgement and then write

the next part(s) in the same manner, until all the data is sent. A VTAP PRO reader will combine successive writes until it has a valid JSON string.

- The “AuthRequired” attribute is currently always false - this is intended for future use, when more advanced authentication schemes may be added.

2 Web Bluetooth API Example

The Web Bluetooth API provides the ability to connect and interact with Bluetooth Low Energy peripherals from JavaScript running within a web page in a web browser. It is supported by Google Chrome and Microsoft Edge browsers running on Windows and other operating systems that support Bluetooth Low Energy. In some browser/OS combinations Web Bluetooth may need to be enabled as an experimental feature.

A demonstration `BleTap.htm` file is available (contact vtap-support@dotorigin.com to request this). It can be launched in a Chrome or Edge browser on a computer with BLE. The main purpose of this is to demonstrate how to develop an application that uses the VTAP PRO reader's custom GATT service.

Just open the file in Chrome/Edge and choose the Connect button. The browser will present a window with the scan results, listing any BLE devices that support the custom service. (It will not show any other BLE devices). After selecting a VTAP PRO reader from that list, select the Pair button. The computer operating system will typically popup a Window (outside of the web browser) asking you if you want to pair. If a PIN has been set using the `BTServerPin` setting in `config.txt`, then you will need to enter that PIN. Once paired, you will be able to connect from the browser without any interaction from the operating system.

The Output section of the web page shows a trace and any taps. The Response JSON is set with a default action, but can be edited to send your own action response to the VTAP reader.

Each time you tap a pass or tag on the VTAP reader it will display the tap data JSON in the Output window and send the currently set action response JSON.

3 iOS Implementation of BLE GATT

You can demonstrate the operation of the VTAP PRO GATT server with an iPhone or iPad using a free BLE scanner app for iOS. (We have tried BLE Scanner 4.0 from bluepixel technologies, although there are a number of other similar apps that allow BLE device services to be explored).

Use the app to scan for BLE devices and see the name that they advertise. (This will be the VTAP serial number, unless it is overridden by a `BTServerName` setting in the `config.txt` file on your VTAP reader). You can connect to the peripheral device and see its advertisement data and services. You can then explore the custom service (`D0F0`) and view the three custom attributes as described earlier under **VTAP BLE GATT custom service**. The BLE scanner app may not show you the full JSON payload or allow you to easily write a complete JSON action response, but it demonstrates how the BLE server works and what is possible from your own app under iOS.

To implement this ability within your own iOS app, you will use the CoreBluetooth API as described here:

<https://developer.apple.com/documentation/corebluetooth/transferring-data-between-bluetooth-low-energy-devices>

Start by creating a `CBCentralManager` class instance and calling `scanForPeripherals`. You can find devices advertising the service UUID `D0F0`.

You can override the `CBCentralManager` `didDiscover` method, to receive notifications of any matching devices and their advertisement data. The advertisement data will contain the VTAP serial number as the device local name (unless it is overridden by a `BTServerName` setting in the `config.txt` file on your VTAP reader).

When your app finds the VTAP PRO reader that it wants to connect to, it can call the `CBCentralManager` `connect` method and expect your supplied `didConnect` override to be called, receiving a `CBPeripheral` object.

With the `CBPeripheral` object you can call the `discoverServices` method and expect your supplied `didDiscoverServices` method to be called with the single `D0F0`. You can call the `CBPeripheral` `discoverCharacteristics` method to find the exposed characteristics and use the `CBPeripheral` `setNotifyValue` method to receive updates whenever the VTAP receive characteristic (with UUID `D0F2`) value changes. You can then use the `CBPeripheral` `readValueForCharacteristic` and `writeValue:data` for characteristic methods to read the tap data payload JSON blob and, optionally, write the action response JSON blob back to the VTAP PRO reader.

4 About Application Notes

Application Notes address topics of interest to small groups of customers, or topics around the use of a VTAP reader with third-party systems.

The main documents available to support your use of the VTAP100-PRO-BW are the Installation Guide for your VTAP reader model and the VTAP Configuration Guide. You will find the latest versions of these, and other useful information at <https://vtapnfc.com>.

If you need further help do contact us by email to vtap-support@dotorigin.com, or by phone +44 (0) 1428 685861 from Europe and Asia, or +1 (562) 262-9642 from Northern and Latin America.