



Release Notes – VTAP core firmware

Our previous VTAP core firmware release version was v2.2.6.0 from Sept 2024

Our latest VTAP core firmware release version is v2.2.7.0 from Jan 2025

We always recommend using the latest core firmware release version, to benefit from the full set of VTAP features and any fixes to date. The previous release version remains available, in case you find that you need to revert for any reason.

Some features can only be used on certain hardware models, or in certain situations. Please check whether you have the right hardware to take advantage of new features.

For more details, or help downloading the latest firmware, please consult the VTAP Configuration Guide, or the VTAP Commands Reference Guide. You will find the latest firmware and guides at <https://vtapnfc.com>

If you have other questions about the latest hardware and firmware capabilities, unit pricing and availability please email us at vtap-sales@dotorigin.com

In summary, the changes in release v2.2.7.0 (Jan 2025) are:

- **New command to return VTAP reader status information as a multi-line JSON string.** New `?info` command to suit VTAP integration developers.
- Improvements such as additional diagnostics and debug options, improved NFC Type 4 card reading error handling, improved card emulation and changing the order in which read outputs are sent to interfaces, with the least time sensitive (keyboard) now last. Firmware size has also been reduced.
- Fixes such as correcting the source reporting a "G" type in response to the `?tap` command and fixes to OSDP.

In summary, the changes in release v2.2.6.0 (Sept 2024) are:

- **Reduced vtapware.dat file size**, improving update speeds.
- **Option to return an error payload when a pass or card read fails**, using `NFCReportReadError`, making it possible to trigger alternative actions when an inappropriate card is presented.
- **Option to disable the USB keyboard device** function of a VTAP reader, using `KBEnable`, which can be useful in Android USB serial integrations.

- **Added Wiegand parity calculation** to increase compatibility with access control systems, using `PassWiegandParity` and `TagWiegandParity` commands.
 - Improvements such as OSDP responses now using a matching 8-bit checksum style where that is used in the request.
 - Fixes such as adding the missing "v" in the firmware version reported in `boot.txt` (on VTAPI00 v4 hardware only).
-

In summary, the changes in release v2.2.5.0 (June 2024) are:

- **Added support for serial barcode/QR scanners**, in addition to the support for Bluetooth barcode/QR scanners already available on VTAPI00-PRO-BW readers.
 - **Added support for tunes and simple MIDI files** to be played through the buzzer. These are configured through a text file controlling buzzer frequency and duration.
 - **Reading NFC type 5 block data** is now possible, in addition to UID reading, which was already available.
 - Fixed issue affecting Google Smart Tap use on Samsung mobile phones.
 - Fixed communications timing issue affecting some VTAPI00-PAC-485 readers.
-

In summary, the changes in release v2.2.4.0 (April 2024) were:

- **Added support for MIFARE Ultralight AES cards and tags** increasing the wide range of physical RFID and NFC technologies now supported, and including the ability to read AES authenticated secure block data.
 - **Added support for DESFire secure read with key diversification following NXP AN10922**
 - **Multiple passive mode interfaces can now read cached tap data** which means that the data from a single pass or card tap can be queried and read by all three passive mode interfaces.
 - **Settings added to permit a beep or LED sequence on startup** - `StartBeep` and `StartLED` settings control this behaviour.
 - **Relay control settings added (for VTAP PRO readers with the VTAP PRO I/O expansion board)** - `PassRelay` and `TagRelay` will operate a relay following a pass, card or tag read.
-

In summary, the changes in release v2.2.3.2 (February 2024) were:

- **Added support for Open Supervised Device Protocol (OSDP)** OSDP defines a method for secure communication between an Access Control Unit (ACU) and all its Peripheral Devices (PDs), such as the VTAP reader. The VTAP reader can now support both Basic and Secure OSDP profiles on VTAPI00 v5 or VTAP50 v2 hardware, as implemented on a range of different access controllers, which typically use an RS-485 interface. This means the new feature will be of most interest if you have a VTAPI00-PAC-485 reader, but OSDP on VTAP readers has been

implemented for use over any serial interface for added flexibility.

- **Number of application key slots increased from 4 to 9** Several VTAP reader operations such as handling DESFire data or using OSDP make use of encryption keys uploaded to app key slots. Up to 9 of these can now be stored securely inside the VTAP reader, along side the 6 ECC key slots that are used for NFC Wallet decryption. All settings that previously required key data in `config.txt` now also permit the key to be uploaded and referenced by key slot instead, for example using `TagReadKeySlot` for MIFARE Classic block reading.
- **Added support for both RGB and GRB serial LEDs** through the new setting `LEDSerialRGB` and `LEDSerialGRB`.
- **Improved compatibility with various PoS devices, including Square terminal**
- **Added several new serial commands** - `?tap` to set data in the tag cache for reading with the passive `?r` command; `?getserial` to read the VTAP reader's serial number, if set; and `?reformat` to reformat the VTAP file system.
- Improved handling of PC sleep-wakeup cycles, which could sometimes interfere with operations over USB interface.
- Improved handling of Wiegand outputs with an odd number of hex digits, using `WiegandPaddingMode` to choose how the padding should work.
- Improved handling of tag and card reading, using `TagByteOrderTypes` to control byte reversal separately for each NFC card or tag type.
- Improved Zmodem file transfer behaviour in the event of a power failure during transfer, which could result in loss of free space on the VTAP file system

In summary, the changes in release v2.2.2.1 (October 2023) were:

- **Increased flexibility in reading DESFire data** that will allow up to six different values to be read from separate files and or applications, and be output together. This uses all the existing DESFire settings, with a number from 1 to 6 now identifying the read item to which each setting relates. In this way each read can be specified separately and linked with the appropriate key.
- **New option to control mass storage access to VTAP readers.** In addition to the existing software lock and hardware lock, there is a new setting `MassStorageEnable` used in the `config.txt` file, which allows a remote host to completely remove or restore mass storage access to VTAP readers without the need for changes to jumpers on the hardware, which may no longer be readily accessible.
- **Added keyboard language translation.** This improves predictability of VTAP keyboard output when using different keyboard language settings on the host computer. The VTAP keyboard output emulates a US keyboard by default. When the host operating system is using another keyboard language, some key presses from the VTAP will be interpreted differently by the host computer. The new option is to provide a keyboard map, which lists key codes corresponding to particular characters, in order to translate to the keyboard language setting used by the host computer.
- Fixed byte reversal issue when sending decimal IDs over the Wiegand

- Fixed occasional Smart Tap payload truncation issue
 - Fixed multiple block NFC Type 4 NDEF record reading issue
-

In summary, the changes in release v2.2.0.2 (August 2023) were:

- **Added support for a new NFC card/tag emulation mode** – operating as an NFC Type 4 tag with dynamic NDEF encoding capabilities and including a smart tag write-back feature
 - **Added information in boot.txt to show loaded app keys** in addition to ECC keys
 - **Added commands to return the VAS/Smart Tap pass type index and key slot used** when reading a pass
 - **Added options to specify the NFC Type 4 NDEF application ID** when reading a tag or HCE device
 - Fixed erroneous UID output for NFC Type B cards/tags
 - Fixed occasional reboot after many serial RGB LED sequences have been played
 - **Added support for Apple ECP2 protocol and Apple Access** – please contact us for details
 - **Added support for new Dot Origin Key-ID credential formats** – please contact us for details
 - **Added support for Smart Tap Generic Private passes** added to Google Wallet directly from an Android app
 - Removed legacy `log.txt` file and associated functions
-

In summary, the changes in release v2.1.12.7 (May 2023) were:

- **Consolidated VTAP firmware** – This release ensures that the exact same features are supported on all versions of the VTAP50 and VTAP100, so far as hardware permits.
 - **Add support for filtering of random UIDs** – A new `IgnoreRandomUID` setting can be used to filter out NFC Type 4 tag reads which are flagged as random, and which can be associated with display-off devices such as Android phones rather than physical cards.
 - **Improved handling of different LED options, settings, and functionality**
 - **Fixed issue with long prefix and postfix strings**
 - **Fixed issue with reading DESFire EV2 cards**
-

Before May 2023 there were **separate firmware releases for VTAP50 and VTAP100 hardware**. The following releases are all marked as 'for VTAP50 only' or 'for VTAP100 only' and listed in chronological order.

In summary, the changes in release v1.1.12.4 (March 2023) for VTAP100 only were:

- **Add support for filtering of random UIDs** – A new `IgnoreRandomUID` setting can be used to filter out NFC Type 4 tag reads which are flagged as random, and which can be associated with display-off devices such as Android phones rather than physical cards.
-

In summary, the changes in release v2.1.12.3 (January 2023) for VTAP50 only were:

- **Firmware file renamed** – New firmware files for VTAP50 are now called `vtapware.dat`, rather than `firmware.dat`, to make it easier to differentiate between files for different models. The filename `firmware.dat` will continue to be accepted on the VTAP50 for backward compatibility.
 - **ZModem improvements** – Additional features to maintain ZModem connections in a wider range of real-world environments.
-

In summary, the changes in release v2.1.12.0 (November 2022) for VTAP50 only were:

- **Added support for reading UIDs from NFC Forum Type 5 cards** – UID can be read with the `NFCType5=U` setting.
 - **Startup delay of 1 second implemented** – the default value of `StartupDelayMS` is now 1000ms to ensure reliable startup in all conditions.
-

In summary, the changes in release v2.1.11.2 (October 2022) for VTAP50 only were:

- **Added support for serial LEDs** – controlled through an `leds.ini` file and new `config.txt` settings.
-

In summary, the changes in release v1.1.12.1 (October 2022) for VTAP100 only were:

- **Added support for reading UIDs from NFC Forum Type 5 cards** – UID can be read with the `NFCType5=U` setting.
 - **Startup delay of 1 second implemented** – the default value of `StartupDelayMS` is now 1000ms to ensure reliable startup in all conditions.
 - **Extra support for Wiegand inputs** – use `WiegandInputEnable` to enable or disable the signals driving red/green LED and/or buzzer.
-

In summary, the changes in release v1.1.11.0 (August 2022) for VTAP100 only were:

- **Improved handling of long command strings over serial**
 - **Additional test and diagnostic commands**
-

In summary, the changes in release v2.1.10.2 (July 2022) for VTAP50 only were:

- **VTAP100 equivalent release** – Includes all the functionality of VTAP100 v1.1.10.2 firmware, where supported by the VTAP50 hardware.
-

In summary, the changes in release v1.1.10.2 (July 2022) for VTAP100 only were:

- **Added support for NDEF data on FeliCa NFC Type 3 tags** – data can be read and decoded with the `NFCType3=N` setting.
 - **Support for latest Google SmartTap generic pass type** – to escape the confines of the basic Google SmartTap use cases.
-

In summary, the changes in release v1.1.10.0 (July 2022) for VTAP100 only were:

- **Added support for DESFire 3DES authentication** – a new setting `DESFireCrypto=1` has been added.
 - **Enhanced support for sending ASCII card and tag data over Wiegand** – a new setting `TagWiegandASCIIFormat` to manipulate ASCII card data as either decimal or hex data.
-

In summary, the changes in release v1.1.9.1 (June 2022) for VTAP100 only were:

- **Improved Zmodem file transfer** – a VTAP firmware update is now five times faster, typically 30 seconds at 115200 baud over RS-485.
 - **All serial communications interfaces enabled by default** – ensures that serial communications are always possible even if the configuration file `config.txt` is missing or damaged.
 - **Uploading key files improved** – key files are only consumed at startup or when a `?k` command is sent over a serial interface. The `?k` command unmounts the USB mass storage to force a directory refresh.
-

In summary, the changes in release v1.1.8.0 (May 2022) for VTAP100 only were:

- **Padding pass data to match format expected by legacy systems** – can now use `TagReadMinDigits` settings to automatically add padding digits to deliver a 32bit or 56bit decimal UID.
-

In summary, the changes in release v1.1.7.0 (April 2022) for VTAP100 only were:

- **Added support for 64 bit decimal decoding of UID or block data** – where the previous limit was 32 bit.
 - **Enhanced UID formatting** – can now use `TagReadOffset` and `TagReadLength` settings to truncate or choose part of a UID.
-

The changes in release v1.1.6.0 (March 2022) for VTAPI00 only were:

- **Facility to automatically search for the correct private key with Apple passes** – by setting `VAS#KeySlot=0` the VTAPI00 will find and use the correct key, if present in its key store.
 - **New features to dynamically enable and disable different/multiple pass types** – the `VASDefaultPassesEnabled` or `STDefaultPassesEnabled` setting allows the user to include up to six Apple and Google passes in the configuration file, select a sub-set that will work at start-up, and then send commands over the serial interface to dynamically change which of those passes it will try to read. This depends on Apple and Google operating/UI restrictions.
 - **New power management feature** – use the `NFCDefaultEnable` setting and `?NFC` serial command to dynamically enable/disable the NFC reader/field of the VTAPI00.
 - **Changes to default serial command settings** – including `COMPortEnable`, `COMPortMode`, `SerialMode`, `Serial2Mode` and `CommandInterfaces` to enable all ports and interfaces by default. This mitigates the risk of a user being unable to connect to their unit if the config is lost.
 - **Command to retrieve public key hash** – using `?h` will return public key data which can help the user confirm which keys have been uploaded.
-

The changes in release v1.1.4.0 (February 2022) for VTAPI00 only were:

- **Support for reading DESFire secure data** – added facility to read DESFire EV1 AES secure data using the setting `NFCType4=D`. This includes the ability to upload up to 4 application keys as `appkey#.txt` files and a set of new related configuration settings prefixed `DESFire...`. The function either decode the 26bit format used by Dot Origin's Key-ID Encoder software, or can return the file data with no format decoding.
-

The changes in release v1.1.3.0 (January 2022) for VTAPI00 only were:

- **Multiple data reading options for each card/tag type** – There is a priority order for reading data types: 1. mobile pass, 2. DESFire secure, 3. Block data, 4. NDEF then 5. UID. You can now choose that if, for instance, reading an NDEF records fails, a UID can be returned.
-

The changes in release v1.1.2.1 (November 2021) for VTAPI00 only were:

- **Changing configuration over serial interfaces** – added a command to send any valid configuration setting over a serial command interface.
-

The changes in release v1.1.1.0 (September 2021) for VTAPI00 only were:

- **Support for RS-485 expansion board** – added RS-485 support via the serial expansion interface to support new hardware options.
- **Improved operation with external power supplies** – added `StartupDelayMS` setting to delay completion of the startup process as a way to avoid potential file system corruption issues.

- **Passive interfaces enabled from start up** – added a facility to start any interface in passive mode at start up using the `PassiveInterfaces` setting.
 - **Improved UID manipulation** – added `TagByteOrder` setting to reverse tag/card UID or block data byte order.
 - **Hardware and software locks** – the action of the hardware lock jumper has changed so that the VTAPI00 configuration is not fully locked by use of the jumper. Instead, the VTAPI00 will no longer be detected as a USB mass storage device, so that its configuration cannot be easily viewed or changed but can still be altered over its serial interfaces. The `boot.txt` message now reports whether both the hardware and/or software locks are set.
 - **LED changes** – a new `LEDSelect` setting allows you to choose which LEDs are enabled, to suit both the VTAPI00 square case and compact case alternatives.
-

The changes in release v1.1.0.0 (May 2021) for VTAPI00 only were:

- **Command interface** – set one or more VTAPI00 `config.txt` options remotely, or retrieve the current settings. You do this by sending commands over any serial interface. Use `$<option>=<value>` to set a command or `%<option>` to read a current setting.
 - **Transfer files using Zmodem** – replace entire files in the VTAPI00 remotely, using the Zmodem file transfer protocol. This can be used for updating firmware, loading private key files, updating `config.txt` or `command.txt`. Send files over any serial interface, using the command `!<filename>`.
 - **Send messages from one interface to another** – send virtual pass or tag payload information to the keyboard, Wiegand, COMPort or other serial interface. Use the syntax `>interface:type:message`. Use prefix and postfix options available on all interfaces to adjust that payload when needed.
 - **Greater control over LED and buzzer** – trigger LED and buzzer actions remotely, by sending `?LED` or `?BEEP` commands over any serial interface. You can also set the LED to flash a specific RGB colour.
 - **Software lock and unlock** – lock or unlock the software remotely with a password, by sending commands over any serial interface. When the VTAPI00 is locked, nothing can be read from or written to the file system. Use the commands `?l` or `?u`.
 - **NDEF reading/decoding for NFC Type 4 tags** – in addition to the Type 2 tags already supported.
-

The changes in release v1.0.4.0 (January 2021) for VTAPI00 only were:

- **Additional tag/card decoding** – use the `MIFAREClassic` option to choose whether to read MIFARE tag/card UIDs or memory. Use a set of new commands prefixed `TagRead...` to identify which blocks to read, and set any key needed to decode these payloads on MIFARE Classic or NFC tags/cards.

- **Identification of type of pass or tag/card read** – the type can be identified as an Apple VAS pass, Google Smart Tap pass, NFC Forum type 1, 2, 3 or 4 , or MIFARE Classic. The type of the last pass read can be queried over the COMport interface, and can be included with the pass data sent over the Wiegand interface.
 - **Additional options for controlling the Wiegand interface** – new commands prefixed `PassWiegand...` or `TagWiegand...` to specify numbers of bits to output over the Wiegand interface, add padding in lieu of parity bits, or interpret the pass payload as a decimal or hexadecimal number.
-

The changes in release v1.0.3.0 (September 2020) for VTAP100 only were:

- **Extra options for extracting NFC card/tag data** – use options prefixed `NDEFTag...` to extract only text records or to extract named items.
-

The changes in release v1.0.2.0 (August 2020) for VTAP100 only were:

- **Additional options for controlling the Wiegand and Serial interfaces** – new commands added to identify which data to read and send over these interfaces. And the `WiegandStartup` command to define data to be sent on startup, before any pass reading.