

---

# **BOLOS Python Loader**

***Release 0.1.30***

**Jul 25, 2023**



---

## Contents

---

<b>1</b>	<b>Scripts</b>	<b>3</b>
<b>2</b>	<b>Script Reference</b>	<b>5</b>



The BOLOS Python loader is a Python library and collection of scripts for interfacing with and managing BOLOS devices from a host computer. See the [Python loader GitHub repository](#) for download and installation instructions.



# CHAPTER 1

---

## Scripts

---

The Python loader includes a collection of useful scripts for managing BOLOS devices. This section includes an overview of some of the most important scripts and how they can be used.

In order to use any of these scripts, the device must be in the dashboard application (no apps are open, the device should display a list of installed apps).

Here is an example using the *deleteApp.py* script from the command-line:

```
python -m ledgerblue.deleteApp --targetId 0x31100002 --appName "Hello World"
```

The above command will delete the app named “Hello World” from the connected Leger Nano S.

See the *Script Reference* for the detailed documentation about each script.





## 2.1 checkGenuine.py

Use attestation to determine if the device is a genuine Ledger device.

```
usage: python -m ledgerblue.checkGenuine [-h] [--targetId TARGETID]
                                           [--issuerKey ISSUERKEY] [--apdu]
```

### 2.1.1 Named Arguments

<b>--targetId</b>	The device's target ID (default is Ledger Blue) Default: 822083586
<b>--issuerKey</b>	Issuer key (hex encoded, default is batch 1) Default: "0490f5c9d15a0134bb019d2afd0bf297149738459706e7ac5be4abc350a1f818057224fce12ec9a6"
<b>--apdu</b>	Display APDU log Default: False

## 2.2 deleteApp.py

Delete the app with the specified name.

```
usage: python -m ledgerblue.deleteApp [-h] [--targetId TARGETID]
                                       [--appName APPNAME] [--appHash APPHASH]
                                       [--rootPrivateKey ROOTPRIVATEKEY]
                                       [--apdu] [--deployLegacy]
                                       [--offline OFFLINE]
```

### 2.2.1 Named Arguments

<b>--targetId</b>	The device's target ID (default is Ledger Blue) Default: 822083586
<b>--appName</b>	The name of the application to delete
<b>--appHash</b>	Set the application hash
<b>--rootPrivateKey</b>	A private key used to establish a Secure Channel (hex encoded)
<b>--apdu</b>	Display APDU log Default: False
<b>--deployLegacy</b>	Use legacy deployment API Default: False
<b>--offline</b>	Request to only output application load APDUs into given filename

## 2.3 derivePassphrase.py

Set a BIP 39 passphrase on the device.

```
usage: python -m ledgerblue.derivePassphrase [-h] [--persistent]
```

### 2.3.1 Named Arguments

<b>--persistent</b>	Persist passphrase as secondary PIN (otherwise, it's set as a temporary passphrase) Default: False
---------------------	---

## 2.4 endorsementSetupLedger.py

Generate an attestation keypair, using Ledger to sign the Owner certificate.

```
usage: python -m ledgerblue.endorsementSetupLedger [-h] [--url URL]
                                                    [--bypass-ssl-check]
                                                    [--apdu] [--perso PERSO]
                                                    [--endorsement ENDORSEMENT]
                                                    [--targetId TARGETID] --key
                                                    {1,2}
```

### 2.4.1 Named Arguments

<b>--url</b>	Server URL Default: "https://hsmprod.hardwarewallet.com/hsm/process"
<b>--bypass-ssl-check</b>	Keep going even if remote certificate verification fails Default: False

<b>--apdu</b>	Display APDU log Default: False
<b>--perso</b>	A reference to the personalization key; this is a reference to the specific Issuer keypair used by Ledger to sign the device's Issuer Certificate Default: "perso_11"
<b>--endorsement</b>	A reference to the endorsement key to use; this is a reference to the specific Owner keypair to be used by Ledger to sign the Owner Certificate Default: "attest_1"
<b>--targetId</b>	The device's target ID (default is Ledger Blue)
<b>--key</b>	Possible choices: 1, 2 Which endorsement scheme to use

## 2.5 endorsementSetup.py

Generate an attestation keypair, using the provided Owner private key to sign the Owner Certificate.

```
usage: python -m ledgerblue.endorsementSetup [-h] --key {1,2}
                                           [--certificate CERTIFICATE]
                                           [--privateKey PRIVATEKEY]
                                           [--targetId TARGETID]
                                           [--issuerKey ISSUERKEY]
                                           [--rootPrivateKey ROOTPRIVATEKEY]
                                           [--apdu]
```

### 2.5.1 Named Arguments

<b>--key</b>	Possible choices: 1, 2 Which endorsement scheme to use
<b>--certificate</b>	Optional certificate to store if finalizing the endorsement (hex encoded), if no private key is specified
<b>--privateKey</b>	Optional private key to use to create a test certificate (hex encoded), if no certificate is specified
<b>--targetId</b>	The device's target ID (default is Ledger Blue) Default: 822083586
<b>--issuerKey</b>	Issuer key (hex encoded, default is batch 1) Default: "0490f5c9d15a0134bb019d2afd0bf297149738459706e7ac5be4abc350a1f818057224fce12ec9a6"
<b>--rootPrivateKey</b>	SCP Host private key
<b>--apdu</b>	Display APDU log Default: False

## 2.6 genCAPair.py

Generate a Custom CA public-private keypair and print it to console.

```
usage: python -m ledgerblue.genCAPair [-h]
```

## 2.7 hashApp.py

Calculate an application hash from the application's hex file.

```
usage: python -m ledgerblue.hashApp [-h] --hex HEX [--targetId TARGETID]
                                     [--targetVersion TARGETVERSION]
```

### 2.7.1 Named Arguments

<b>--hex</b>	The application hex file to be hashed
<b>--targetId</b>	The device's target ID (default is Ledger Blue)
<b>--targetVersion</b>	Set the chip target version

## 2.8 hostOnboard.py

**Warning:** Using this script undermines the security of the device. Caveat emptor.

```
usage: python -m ledgerblue.hostOnboard [-h] [--apdu] --id {0,1,2} [--pin PIN]
                                         [--prefix PREFIX]
                                         [--passphrase PASSPHRASE]
                                         [--words WORDS]
```

### 2.8.1 Named Arguments

<b>--apdu</b>	Display APDU log Default: False
<b>--id</b>	Possible choices: 0, 1, 2 Identity to initialize
<b>--pin</b>	Set a PINs to backup the seed for future use
<b>--prefix</b>	Derivation prefix
<b>--passphrase</b>	Derivation passphrase
<b>--words</b>	Derivation phrase

## 2.9 listApps.py

List all apps on the device.

```
usage: python -m ledgerblue.listApps [-h] [--targetId TARGETID]
                                     [--rootPrivateKey ROOTPRIVATEKEY]
                                     [--apdu] [--deployLegacy] [--scp]
```

### 2.9.1 Named Arguments

<b>--targetId</b>	The device's target ID (default is Ledger Blue) Default: 822083586
<b>--rootPrivateKey</b>	The Signer private key used to establish a Secure Channel (otherwise, a random one will be generated)
<b>--apdu</b>	Display APDU log Default: False
<b>--deployLegacy</b>	Use legacy deployment API Default: False
<b>--scp</b>	Use a secure channel to list applications Default: False

## 2.10 loadApp.py

Load an app onto the device from a hex file.

```
usage: python -m ledgerblue.loadApp [-h] [--targetId TARGETID]
                                     [--targetVersion TARGETVERSION]
                                     [--apiLevel APILEVEL] --fileName FILENAME
                                     [--icon ICON] [--curve CURVE]
                                     [--path PATH] [--path_slip21 PATH_SLIP21]
                                     --appName APPNAME [--signature SIGNATURE]
                                     [--signApp] [--appFlags APPFLAGS]
                                     [--bootAddr BOOTADDR]
                                     [--rootPrivateKey ROOTPRIVATEKEY]
                                     [--signPrivateKey SIGNPRIVATEKEY] [--apdu]
                                     [--deployLegacy] [--delete] [--params]
                                     [--tlv] [--dataSize DATASIZE]
                                     [--appVersion APPVERSION]
                                     [--offline OFFLINE] [--offlineText]
                                     [--installparamsSize INSTALLPARAMSSIZE]
                                     [--tlvraw TLVRAW] [--dep DEP] [--nocrc]
```

### 2.10.1 Named Arguments

<b>--targetId</b>	The device's target ID (default is Ledger Blue) Default: 822083586
-------------------	---

<b>--targetVersion</b>	Set the chip target version
<b>--apiLevel</b>	Set the API level of the SDK used to build the app Default: -1
<b>--fileName</b>	The application hex file to be loaded onto the device
<b>--icon</b>	The icon content to use (hex encoded)
<b>--curve</b>	A curve on which BIP 32 derivation is locked (“secp256k1”, “secp256r1”, “ed25519” or “bls12381g1”), can be repeated
<b>--path</b>	A BIP 32 path to which derivation is locked (format decimal a’/b’/c), can be repeated
<b>--path_slip21</b>	A SLIP 21 path to which derivation is locked
<b>--appName</b>	The name to give the application after loading it
<b>--signature</b>	A signature of the application (hex encoded)
<b>--signApp</b>	Sign application with provided signPrivateKey Default: False
<b>--appFlags</b>	The application flags Default: 0
<b>--bootAddr</b>	The application’s boot address
<b>--rootPrivateKey</b>	The Signer private key used to establish a Secure Channel (otherwise a random one will be generated)
<b>--signPrivateKey</b>	Set the private key used to sign the loaded app
<b>--apdu</b>	Display APDU log Default: False
<b>--deployLegacy</b>	Use legacy deployment API Default: False
<b>--delete</b>	Delete the app with the same name before loading the provided one Default: False
<b>--params</b>	Store icon and install parameters in a parameter section before the code Default: False
<b>--tlv</b>	Use install parameters for all variable length parameters Default: False
<b>--dataSize</b>	The code section’s size in the provided hex file (to separate data from code, if not provided the whole allocated NVRAM section for the application will remain readonly).
<b>--appVersion</b>	The application version (as a string)
<b>--offline</b>	Request to only output application load APDUs into given filename
<b>--offlineText</b>	Request to only output application load APDUs into given filename in text mode Default: False

<b>--installparamsSize</b>	The loaded install parameters section size (when parameters are already included within the .hex file.
<b>--tlvraw</b>	Add a custom install param with the hextag:hexvalue encoding
<b>--dep</b>	Add a dependency over an appname[:appversion]
<b>--nocrc</b>	Skip CRC generation when loading Default: False

## 2.11 loadMCU.py

Load the firmware onto the MCU. The MCU must already be in bootloader mode.

```
usage: python -m ledgerblue.loadMCU [-h] --targetId TARGETID --fileName
                                     FILENAME [--bootAddr BOOTADDR] [--apdu]
                                     [--reverse] [--nocrc]
```

### 2.11.1 Named Arguments

<b>--targetId</b>	The device's target ID
<b>--fileName</b>	The name of the firmware file to load
<b>--bootAddr</b>	The firmware's boot address
<b>--apdu</b>	Display APDU log Default: False
<b>--reverse</b>	Load HEX file in reverse from the highest address to the lowest Default: False
<b>--nocrc</b>	Load HEX file without checking CRC of loaded sections Default: False

## 2.12 mcuBootloader.py

Request the MCU to execute its bootloader.

```
usage: python -m ledgerblue.mcuBootloader [-h] [--targetId TARGETID]
                                           [--rootPrivateKey ROOTPRIVATEKEY]
                                           [--apdu]
```

### 2.12.1 Named Arguments

<b>--targetId</b>	The device's target ID (default is Ledger Blue) Default: 822083586
<b>--rootPrivateKey</b>	The Signer private key used to establish a Secure Channel (otherwise a random one will be generated)

**--apdu**                      Display APDU log  
                              Default: False

## 2.13 resetCustomCA.py

Remove all Custom CA public keys previously enrolled onto the device.

```
usage: python -m ledgerblue.resetCustomCA [-h] [--targetId TARGETID] [--apdu]
                                           [--rootPrivateKey ROOTPRIVATEKEY]
```

### 2.13.1 Named Arguments

**--targetId**                The device's target ID (default is Ledger Blue)  
                              Default: 822083586

**--apdu**                    Display APDU log  
                              Default: False

**--rootPrivateKey**        The Signer private key used to establish a Secure Channel (otherwise a random one will be generated)

## 2.14 runApp.py

```
usage: python -m ledgerblue.runApp [-h] [--targetId TARGETID] [--apdu]
                                   [--rootPrivateKey ROOTPRIVATEKEY] --appName
                                   APPNAME
```

### 2.14.1 Named Arguments

**--targetId**                The device's target ID (default is Ledger Blue)  
                              Default: 822083586

**--apdu**                    Display APDU log  
                              Default: False

**--rootPrivateKey**        The Signer private key used to establish a Secure Channel (otherwise a random one will be generated)

**--appName**                The name of the application to run

## 2.15 runScript.py

Read a sequence of command APDUs from a file and send them to the device. The file must be formatted as hex, with one CAPDU per line.



```
usage: python -m ledgerblue.runScript [-h] [--fileName FILENAME] [--apdu]
                                     [--scp] [--targetId TARGETID]
                                     [--elfFile ELFFILE]
                                     [--rootPrivateKey ROOTPRIVATEKEY]
```

### 2.15.1 Named Arguments

<b>--fileName</b>	The name of the APDU script to load
<b>--apdu</b>	Display APDU log Default: False
<b>--scp</b>	Open a Secure Channel to exchange APDU Default: False
<b>--targetId</b>	The device's target ID (default is Ledger Nano S). If --elfFile is used, the targetId from the ELF file will be used instead. Default: 823132162
<b>--elfFile</b>	ELF file from which the target ID is fetched. Overrides '--targetId'
<b>--rootPrivateKey</b>	The Signer private key used to establish a Secure Channel (otherwise a random one will be generated)

## 2.16 setupCustomCA.py

Enroll a Custom CA public key onto the device.

```
usage: python -m ledgerblue.setupCustomCA [-h] [--targetId TARGETID] [--apdu]
                                           [--rootPrivateKey ROOTPRIVATEKEY]
                                           --public PUBLIC --name NAME
```

### 2.16.1 Named Arguments

<b>--targetId</b>	The device's target ID (default is Ledger Blue) Default: 822083586
<b>--apdu</b>	Display APDU log Default: False
<b>--rootPrivateKey</b>	The Signer private key used to establish a Secure Channel (otherwise a random one will be generated)
<b>--public</b>	The Custom CA public key to be enrolled (hex encoded)
<b>--name</b>	The name to assign to the Custom CA (this will be displayed on screen upon auth requests)

## 2.17 signApp.py

## 2.18 updateFirmware.py

```
usage: python -m ledgerblue.updateFirmware [-h] [--url URL]
                                           [--bypass-ssl-check] [--apdu]
                                           [--perso PERSO] --firmware FIRMWARE
                                           [--targetId TARGETID] --firmwareKey
                                           FIRMWAREKEY
```

### 2.18.1 Named Arguments

<b>--url</b>	Server URL Default: “ <a href="https://hsmprod.hardwarewallet.com/hsm/process">https://hsmprod.hardwarewallet.com/hsm/process</a> ”
<b>--bypass-ssl-check</b>	Keep going even if remote certificate verification fails Default: False
<b>--apdu</b>	Display APDU log Default: False
<b>--perso</b>	A reference to the personalization key; this is a reference to the specific Issuer keypair used by Ledger to sign the device’s Issuer Certificate Default: “perso_11”
<b>--firmware</b>	A reference to the firmware to load
<b>--targetId</b>	The device’s target ID (default is Ledger Blue) Default: 822083586
<b>--firmwareKey</b>	A reference to the firmware key to use

## 2.19 verifyApp.py

```
usage: python -m ledgerblue.verifyApp [-h] --hex HEX --key KEY --signature
                                       SIGNATURE
```

### 2.19.1 Named Arguments

<b>--hex</b>	The hex file of the signed application
<b>--key</b>	The Custom CA public key with which to verify the signature (hex encoded)
<b>--signature</b>	The signature to be verified (hex encoded)

## 2.20 verifyEndorsement1.py

Verify a message signature created with Endorsement Scheme #1.

```
usage: python -m ledgerblue.verifyEndorsement1 [-h] --key KEY --codehash
CODEHASH --message MESSAGE
--signature SIGNATURE
```

### 2.20.1 Named Arguments

<b>--key</b>	The endorsement public key with which to verify the signature (hex encoded)
<b>--codehash</b>	The hash of the app associated with the endorsement request (hex encoded)
<b>--message</b>	The message associated to the endorsement request (hex encoded)
<b>--signature</b>	The signature to be verified (hex encoded)

## 2.21 verifyEndorsement2.py

Verify a message signature created with Endorsement Scheme #2.

```
usage: python -m ledgerblue.verifyEndorsement2 [-h] --key KEY --codehash
CODEHASH --message MESSAGE
--signature SIGNATURE
```

### 2.21.1 Named Arguments

<b>--key</b>	The endorsement public key with which to verify the signature (hex encoded)
<b>--codehash</b>	The hash of the app associated with the endorsement request (hex encoded)
<b>--message</b>	The message associated to the endorsement request (hex encoded)
<b>--signature</b>	The signature to be verified (hex encoded)