CANedge2 Intro and Tools

Release FW 01.04.01

Oct 12, 2021

CONTENTS

0.1 CANedge2 - get started

This guide provides a simple intro to each step of your workflow - including software & API tools.

0.1.1 About this manual

0.1.1.1 Purpose

The CANedge2 Intro focuses on the following:

- How to get started with the CANedge2
- How to use relevant software & API tools

The document is structured by the steps you go through when using the device for the first time.

0.1.1.2 Other documentation

The CANedge1 Docs and CANedge2 Docs serve as the product manual. Those docs detail the hardware, configuration and concepts beyond the scope of the CANedge Intro.

0.1.1.3 Notation used

The following notation is used throughout this documentation:

Admonitions

Note: Used to highlight supplementary information

Warning: Used if incorrect use may result in major loss of data and/or time

Danger: Used if incorrect use may result in personal injury or death

0.2 Configure your device

Below we outline how to update your device firmware and configuration.

0.2.1 Check for firmware updates

Before you start, we suggest that you check if a newer firmware exists. You can compare the `fw_ver` in your `device.json` file on the SD card vs. the latest firmware in the CANedge2 Docs.

Note: For `MAJOR`/`MINOR` updates your Configuration File needs to be updated.

0.2.2 Configure your CANedge

1. Extract the device SD card, insert it into your PC and open the config editor tool
2. Load your SD card `config-XX.YY.json` file in the editor via the Configuration File dropdown[1]
3. Modify your config, press "Review changes" and verify your edits
4. Download the Configuration File and copy/paste it to your SD card (overwrite the original file)
5. Safely eject the SD card and re-insert it into your CANedge2

The new Configuration File is now loaded by your device the next time it is powered on.

Note: Optionally open the online editor, right-click and "Save as..." to store the editor for offline use

Note: An invalid Configuration File will be defaulted by the device upon power up

Note: The UIschema lets you toggle advanced configuration settings on/off (off by default)

0.2.3 Configuration tools

0.2.3.1 Editor tools

For the best experience, we recommend to use one of the configuration editors below[1].

Simple config editor (offline/online)

The 'simple editor' lets you load and edit your Configuration File[2]:

Note: Optionally open the online editor, right-click and "Save as..." to store the editor for offline use

[1] The Rule Schema (`schema-XX.YY.json`) determines the structure of the editor, while the Configuration File (`config-XX.YY.json`) contains your actual configuration. The optional UIschema (`uischema-XX.YY.json`) determines the styling of the editor (e.g. for toggling advanced settings). For further details on the JSON Schema concept, see the CANedge Docs.

[1] The Configuration File is a JSON text file. This means you can in principle perform updates via any standard text editor. However, in doing so you will not benefit from the Rule Schema, which ensures that you perform valid edits. For most purposes, we therefore recommend to use a Schema based editor tool.

[2] The CANedge browser tools like the configuration editors (online/offline) work on modern browser like Chrome, Firefox and Internet Edge, but not Internet Explorer.

Over-the-air editor tools

With a CANedge2, you can perform updates over-the-air by changing the device Configuration File on your server. This can be done e.g. within CANcloud or via the OTA batch manager.

0.2.3.2 Using the config editor

Below we provide details on how to use the config editor.

Documentation

We strongly recommend that you review the Configuration section of the CANedge docs. This explains how the JSON Schema concept works incl. the role of the Configuration File, Rule Schema and UIschema. Further, it provides detailed examples for some of the more advanced configuration settings like CAN ID filters.

Presentation mode - simple vs. advanced

The editor tools will by default hide advanced settings for simplicity. To show all the available settings, you can switch the Presentation Mode in the sidebar.

Support tools

The editors add a number of configuration tools in the bottom toolbar:

- **Encryption tool**: This tool helps you encrypt passwords. For details see the CANedge Docs and the encryption tool section. For batch encryption, see the OTA batch manager section
- **Filter checker**: When setting up CAN ID filters, this tool can help evaluate if a given CAN ID will pass through the filter or not. It also provides guidance for setting up J1939 PGN filters
- **Partial config loader**: This lets you load, schema-validate and merge a partial Configuration File (e.g. a transmit list) into your active Configuration File
- **Bit timing calculator**: You can use the "Bit-timing (advanced)" mode to set a custom bit rate for your application and this calculator can be useful in checking your settings
- **Schema & config loader**: This will be open by default when using the editor and lets you load the UIschema, Rule Schema and Configuration File for use in configuration of the device

New major/minor Firmware

If you need to update to a new major/minor CANedge Firmware (e.g. from `00.07.04` to `01.02.04`), you will need to add a matching Configuration File to the device SD card. You can use the config editor to help update an existing Configuration File to a new Firmware structure:

1. Download the new Firmware zip from the CANedge docs
2. Load the new Rule Schema via the editor sidebar Rule Schema dropdown
3. Load your old Configuration File via the editor sidebar Configuration File dropdown
4. Perform updates if needed to ensure validity with the new Rule Schema
5. Review and download your new Configuration File

For details on how to update the Firmware, see the CANedge docs.

0.2.3.3 Encryption tool

The CANedge supports encryption of passwords - see the CANedge Docs for details.

Below we outline how to easily encrypt fields using the encryption tool within the config editor.

Generating keys & encrypting plain text data

Within the config editor, click the "Encryption tool" (lock icon) to open the tool.

In the tool, paste the public key (`kpub`) value from your `device.json` file and click "Create keys"[3].

This produces two keys:

1. **Server public key**: Should be added in the security section of the device Configuration File
2. **Encryption key**: Used for the encryption of plain text data (e.g. passwords)

You are now able to encrypt plain text data as follows:

1. Enter a plain text password and click "Encrypt"
2. Copy the encrypted password to the relevant field in the Configuration File
3. Set the corresponding key format to "Encrypted" in the Configuration File

Re-using an encryption key

You can securely store the encryption key for later use. This lets you use the second mode of the encryption tool to encrypt fields using an existing key.

This way you can later update/add passwords without changing the server public key or any pre-encrypted data in your Configuration File.

Warning: If you use multiple WiFi access points, their passwords must be consistently plain or encrypted

Note: If you need to encrypt passwords on a large number of CANedge2 devices, we recommend using the OTA batch manager tool

[3] You can extract the `kpub` from the `device.json` file on the device SD card

0.3 Record raw data

Below we outline how to record raw data to the device SD card.

0.3.1 Preparation

Before you connect your device, it is important that you do the following:

1. Read the introduction and hardware installation guide in the CANedge2 Docs
2. Verify that the pin-out of your application, adapter cable & CANedge match

0.3.2 Verify that you can log data

1. Connect & power the CANedge in your application via Channel 1 (green LED lights up)
2. Verify that the device records data to the SD card (yellow & red LEDs blink)
3. Disconnect the device, extract the SD and confirm that the `LOG/` folder now contains data[1]

If you're logging data from cars or heavy-duty vehicles, see our OBD2/J1939 sections. If you're having trouble logging data, see our troubleshooting section.

Note: Once you're done testing we recommend that you configure filters, prescalers & compression to reduce your file size (often by 90%+) - see also our tips & tricks section

See the online documentation for more details on logging OBD2 and J1939 data

[1] It is important that you disconnect your device before extracting the SD card to avoid SD card corruption. Similarly, when ejecting the SD card from your PC, make sure to use the 'safe eject' functionality

0.4 Transfer data

0.4.1 Transfer data via SD card

When you're ready to process your data, you'll need to transfer it from the device SD card:

1. Disconnect the logger from your application (the CANedge is 100% power safe)
2. Extract the SD card, insert it into your PC and open the `LOG/` folder
3. Transfer the log files you wish to process to your PC

0.4.2 Transfer data via WiFi

In addition to SD card transfer, the CANedge2 lets you auto-push data to your server via WiFi:

1. Follow the guide to *add a WiFi access point*
2. Follow the guide to *set up your S3 server*[1]
3. Configure the CANedge2 with your WiFi access point and S3 server details
4. Power your device and use *CANcloud* to verify that files are uploaded to your S3 server[2]

Before you deploy your CANedge2 in the field, see our WiFi tips & tricks.

If you're having trouble uploading files, see our troubleshooting section. For technical details, see the CANedge2 Docs. See also the *server tools* - e.g. to *mount your server* as a local drive.

Note: We **strongly recommend** that you test your setup before remote deployment: Ensure that the device logs data, connects to your server and uploads files as expected[3]

Note: When your device is connected to your server with OTA enabled, configuration changes made via the SD will be overwritten by the server Configuration File[4]

0.4.2.1 WiFi access points

To enable WiFi data transfer in station mode, you'll need a WiFi access point:

1. Select a WiFi access point for use with the CANedge2 (see below)
2. If you're using our 3G/4G USB router, follow our *setup guide*
3. Add the WiFi access point SSID (name) and password in your Configuration File
4. Test that the CANedge2 is able to connect to the WiFi access point (blue LED lights up[1])

[1] To test your file upload, you can optionally use our open AWS S3 server

[2] When the device first connects to your server, it will upload the Rule Schema, Configuration File and Device File. When the device has finished recording a new log file to the SD card, this will be automatically pushed to your server as well.

[3] It is important that you test this in the actual location where you are installing your CANedge2. For example, mounting the device behind panels or in a box may block the WiFi access/speed.

[4] Once your device is connected to your server, we recommend to update the Configuration File via your server (e.g. via CANcloud). If you prefer to update it via the SD card e.g. during tests, ensure that you delete the Configuration File from your S3 server device folder before making a change on the SD card (to avoid triggering an OTA update).

[1] The blue LED lights up when the device connects to the WiFi access point. The device will do so when it attempts to upload files or WiFi-sync the real-time clock. If the blue LED is not light up, it can either be because the CANedge is not actively trying to upload files (e.g. if no log files are on the SD card) or if it is unable to connect to the configured WiFi access point.

Access point options

There are multiple options that you can use as WiFi access points, depending on your use case:

Standard WLAN router: In "fixed locations" (warehouses, garages, mines, offices etc.) a WLAN router can be used as an access point. It's popular in use cases where the CANedge2 is installed in a stationary application - or in e.g. garages where vehicles may return periodically.

Cellular router (3G/4G): You can use a 3G/4G router to allow the CANedge2 to upload data e.g. while on-the-road. For example, you can power a USB 3G/4G router via the CANedge2 2nd port. You can also deploy a cellular router to serve as a shared access point for several CANedge2 units that come within range periodically.

Smartphone (3G/4G): A simple option for e.g. test purposes can be a smartphone's hotspot.

USB 3G/4G router setup

Below we briefly outline how to setup our 3G/4G USB router.

Setup your 3G/4G USB router

1. Insert your nano SIM card into the small slot in the USB stick
2. Power the USB hotspot by inserting it into your PC - the blue LED should come on
3. Via your PC, connect to the hotspot SSID `4G-UFI-XXXX` with password `12345678`
4. Access the settings by entering `192.168.0.1` in your browser (`user: admin, password: admin`)
5. If you see "No service", click Connection Settings and add your SIM's PIN code
6. Change the SSID name & pass under Settings/WLAN/WLAN Basic Settings
7. Click the connect button on the Home page and verify that it stays connected[1]
8. Test that you can browse the internet via your laptop when connected via the hotspot
9. Test that you can still connect via the hotspot after power cycling it a few times
10. Use the *editor* to add your new SSID/password in your CANedge2 Configuration File
11. If relevant, turn on the 2nd port 'power out' in the Configuration File[2]
12. Test that the CANedge2 is able to connect to the hotspot and your server[3]

Note: Some SIM card providers will by default block high speed data transfer. If you experience upload speed issues, we recommend contacting your SIM card provider to ensure that you are getting the highest possible upload speed

[1] In some cases it is necessary to update the APN info in the hotspot settings under Profile Management (create a new profile when doing so). After updating your settings, it may be necessary to power cycle the hotspot and wait for 1-2 minutes.

[2] If you're powering the USB hotspot via the CANedge2 2nd port, we recommend to set the power schedule to `From: 00:00` and `To: 23:59`. This ensures the hotspot is power cycled daily, which is recommended.

[3] If you're powering the USB hotspot via the CANedge2 2nd port, the total setup consumes up to 4 W (CANedge2: 1W, hotspot: 1-3W). Practically all in-field applications (cars, trucks, ...) support this, but an office power supply might not. If you experience issues with your upload tests, try powering the hotspot separately (e.g. from a USB adapter or your laptop).

Note for users in USA

For US users, we recommend Soracom as data provider with this hotspot based on experience.

To use Soracom, go to Settings/Dial-up/Profile Management/New Profile.

Here, create a new profile (do not use the existing one) and add the below:

- Profile name: `MySoracomProfile`
- APN: `soracom.io`
- User name: `sora`
- Password: `sora`

As above, test that the hotspot retains internet connection across a number of power cycles.

Regarding log file split size

For 3G/4G transfer, we generally recommend to reduce the log file size to e.g. 5-10 MB (rather than the default 50 MB). The CANedge2 does not currently support 'resumable uploads' so if the transfer fails mid-transfer (e.g. due to temporary lack of coverage), the file transfer will start over when the device regains connectivity. Thus, to minimize wasted data and ensure steady throughput, a lower split size is recommended. Once the data is on your server, you can use various tools to concatenate the log files (e.g. asammdf, our Python API etc).

0.4.2.2 S3 server

Below we outline three types of S3 servers that you can use with your CANedge2:

1. *Local* - upload files to a self-hosted server on a local network (warehouse, boat, office, ...)
2. *Dedicated* - upload files to a self-hosted server over the internet via port forwarding
3. *Cloud* - upload files to a cloud server for scalability, processing power and ease-of-use

You can learn more about S3 buckets & objects in the CANedge Docs.

Note: If you need to upload data via the internet (rather than via a local network) we recommend that you start with an *AWS S3 cloud*. It's simple to setup and practically free at small scale. You can always switch later via an over-the-air update.

Local MinIO server

The CANedge2 can connect to a local server via the local WiFi access point - ideal for e.g. stationary use cases. We recommend MinIO which is the #1 free open source S3 server.

Below we outline how to set up a local MinIO server in 5 minutes.

Setting up a local MinIO server

Below we outline how you can set up a **local server** on your Windows PC:

1. Download the MinIO S3 server (Windows, Linux AMD64)[1]
2. Open the command prompt in the folder, paste in the below and hit enter

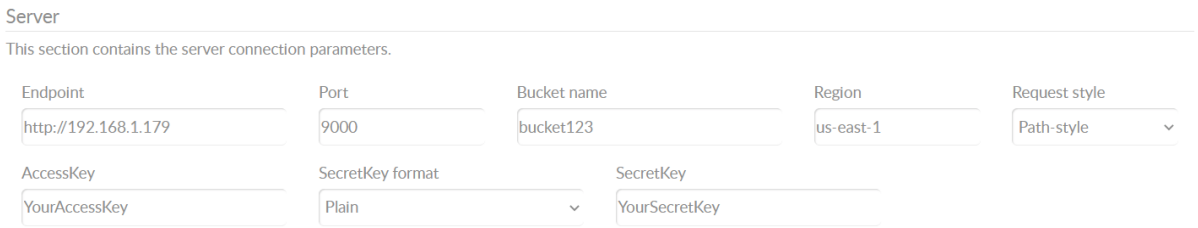
[1] The download links are for release `2021-09-18T18-09-59Z`, which we use as basis for this guide. For later releases and e.g. other Linux builds, see the release page.

```
SET MINIO_ROOT_USER=YourAccessKey
SET MINIO_ROOT_PASSWORD=YourSecretKey
minio.exe server C:\DATA --console-address ":9001"
```

1. Open the 'MinIO console' by entering the *console address* in your browser (port `9001`)
2. Login to the console, go to "Buckets" and create a new bucket (default settings)
3. You now have your endpoint, port (`9000`), bucket and credentials[2]
4. Check if you can login via *CANcloud* on *the host PC* (using Firefox[3])
5. Check if you can login via CANcloud on *another PC* on the network (using Firefox[3])

Example: CANedge2 MinIO S3 (local) server configuration

Below are example Configuration File details for a MinIO S3 server (*editor view* and JSON). Note that the region is not used for MinIO servers and can be set to `us-east-1` for simplicity.




```
"server": {
           "endpoint": "http://192.168.1.179",
           "port": 9000,
           "bucket": "bucket123",
           "region": "us-east-1",
           "request_style": 0,
           "accesskey": "YourAccessKey",
           "keyformat": 0,
           "secretkey": "YourSecretKey"
         }
```

Note: It is strongly recommended that the host PC/system is continuously active to ensure that the self-hosted MinIO S3 server is always available

[2] MinIO provides multiple IP endpoints - for running a local server setup, use the one corresponding to your router's IP structure (see 'Default Gateway' when running ipconfig in the command prompt) - e.g. 192.168.0.178. The port should be `9000` when you log in via CANcloud or configure your CANedge2. Once you're done testing, you can update your SecretKey e.g. via Norton's password generator. You can then update it by using `SET MINIO_ROOT_PASSWORD=YourNewSecretKey`

[3] Chrome recently blocked access to non-TLS endpoints. By default, your MinIO S3 server is setup to run without TLS enabled. This means that if you wish to login to your MinIO S3 server via CANcloud using Chrome, you will need to enable TLS on your MinIO server and CANedge2 devices (as per our separate guide). Since TLS is an advanced topic, we recommend that you initially use Firefox to login via CANcloud, as Firefox does not block access. If you wish to login via Chrome on the *host PC*, you can also download and unzip the latest CANcloud release and open the `index.html` file via Chrome. Finally, you can use Chrome with a non-TLS MinIO server if you have administrative access as outlined here (though this is an advanced topic).

Creating a bat file

If you later need to start this server again, you can optionally create a `*.bat` file with the lines from earlier, which you can then double-click to start your server. This bat file can also be setup to run when your machine boots.

Using the MinIO console

MinIO supports a powerful 'console' view that offers a graphical user interface for controlling your MinIO S3 settings and monitoring your data flow. This can be viewed as a 'server management tool', while CANcloud can be viewed as a 'device & data management tool'.

Using the MinIO client tool

The MinIO Client tool lets you perform various operations on your MinIO server via the command line. It provides a modern alternative to UNIX commands like `ls`, `cat`, `cp`, `mirror`, `diff` etc.

To get started with this tool, follow the below steps:

1. Download the MinIO client
2. Open your command line and enter the below (`<ALIAS>` is a nickname for your server) `mc config host add <ALIAS> <YOUR-S3-ENDPOINT> <YOUR-ACCESS-KEY> <YOUR-SECRET-KEY>`
3. You can now e.g. create a bucket on your server as follows: `mc mb <ALIAS>/<BUCKETNAME>`

Example

Adding a HTTP trace for your MinIO server:

```
mc config host add myserver http://192.168.1.179:9000 YourAccessKey YourSecretKey
mc mb myserver/mybucket
mc admin trace myserver
```

For a full list of MinIO Client commands, see the MinIO Client quickstart.

Dedicated MinIO server

For some use cases, you need to upload data from outside your server's local network.

In this case, you can either use a dedicated server or cloud. Both let the device push data to your server while connected to an external WiFi access point (e.g. a remote WLAN or 3G/4G hotspot).

Warning: If you're new to servers, we recommend that you start with an *AWS S3 cloud*. This way you do not have to consider port forwarding, firewalls etc. Once your setup is in place, you can always switch to a MinIO server later via an over-the-air update

Setting up a dedicated MinIO server (Windows)

To make a local MinIO server dedicated, you can port forward it:

1. Log into your router settings (often via `http://192.168.0.1`)
2. Go to the Port Forwarding section (often under WAN or Advanced)
3. Create a new port forward entry (IPv4)

4. In the 'Local IP', add the local network MinIO endpoint (e.g. `192.168.0.178`)
5. In the local start/end port, add the MinIO endpoint port (e.g. `9000`)
6. Use the same MinIO port for the external start/end port (if relevant)
7. Find your public WAN IP by googling "My IP" (e.g. `176.21.122.154`)
8. Combine your public WAN IP with the MinIO port (e.g. `176.21.122.154:9000`)
9. Verify that you can connect to the server via e.g. your smartphone or other external network

Warning: The basic setup is intended for small-scale use. For use cases involving a larger number of CANedge2 devices, the setup may need to be modified for scalability

Cloud servers (AWS, Google, Azure, . . .)

The CANedge2 can be used with multiple cloud servers for convenience and easy scalability.

AWS S3 cloud server [recommended]

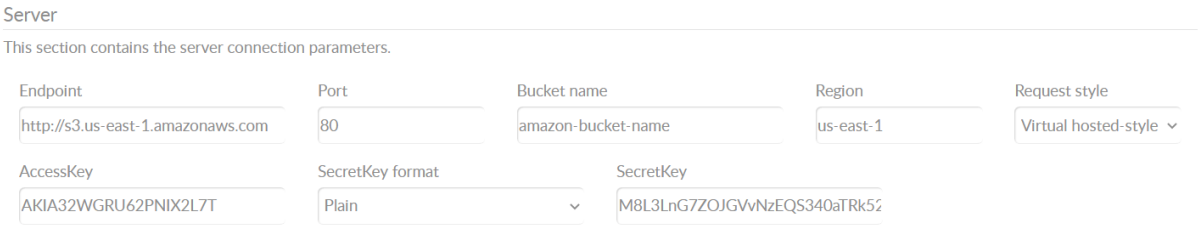
1. Sign up for a free account (requires card details, but no charges are made)
2. Once signed in, search for S3 under 'Services'
3. Under S3, click 'Create bucket' - provide a name and select a nearby region[1]
4. In your bucket go to Permissions/CORS configuration and paste this JSON content
5. Create an AWS IAM user following this guide to get your AccessKey/SecretKey
6. Note your region (e.g. `us-east-2`) and your endpoint (`http://s3.[region].amazonaws.com`)

You can now *configure your CANedge2* and log into *CANcloud* using your details.

Note: When configuring your device for AWS S3, set the request style to `Virtual hosted style`. Also, make sure to use `http://` in your endpoint and set the port to `80`

Example: CANedge2 AWS S3 server configuration

Below are example Configuration File details for an AWS S3 server (*editor view* and JSON):



[1] Selecting a nearby region with low latency is key to ensuring fast data transfer rates.

```
"server": {
    "endpoint": "http://s3.us-east-1.amazonaws.com",
    "port": 80,
    "bucket": "amazon-bucket-name",
    "region": "us-east-1",
    "request_style": 1,
    "accesskey": "AKIA32WGRU62PNIX2L7T",
    "keyformat": 0,
    "secretkey": "M8L3LnG7ZOJGVvNzEQS340aTRk52NS++oQgwr8VV"
  }
```

Google cloud server

Google Cloud Storage supports S3 via 'Interoperability':

See also our video of how to set up your Google Cloud bucket and how to set up CORS.

1. Sign up for a free account (this requires your card details, but no charges are made)
2. Under Storage/Browser select "Create bucket" and select your preferred region
3. Go to Settings/Interoperability, Enable Interoperability and "Create a new key"
4. To enable CORS, start Google Cloud Shell via the `>_` icon in the upper right corner
5. Enter the following command:

   ```
   echo '[{"maxAgeSeconds": 3600, "method": ["GET", "OPTIONS", "HEAD", "PUT", "POST"],
   "origin": ["*"], "responseHeader": ["*"]}]' > cors-config.json
   ```

6. Next, modify below with your bucket name and run it in the shell: `gsutil cors set cors-config.json gs://[YOUR_OWN_BUCKET_NAME]`

You can now configure your CANedge2 and log into CANcloud using the endpoint, http://storage.googleapis.com, your bucket name and the Interoperability storage access keys.

Note: Some S3 API calls are not supported via GCS Interoperability, incl. parts of CANcloud. If you need full support for the API, consider AWS or MinIO

Other S3 cloud servers (Wasabi, DigitalOcean)

You can also use other S3 cloud servers like Wasabi or DigitalOcean. The concept of setting these up is similar to e.g. AWS and hence not described in detail here. If questions, please contact us.

Microsoft Azure cloud server (using MinIO)

Azure does not directly support S3, but you can use MinIO to add the S3 object layer to Azure's accounts & blobs structure[2]. See the MinIO-Azure gateway quickstart guide. For production deployments, see the MinIO-Azure marketplace guide.

[2] Note that Azure + MinIO is a higher-latency solution vs. the native S3 cloud servers. If your use case requires fast WiFi data transfer (e.g. for periodic uploads), consider one of the other server options. Note also that you can host MinIO on your Azure server directly.

Connect to your S3 server via HTTPS

The CANedge2 lets you upload data via HTTP or HTTPS. HTTP is simpler to setup/maintain, but if security is a priority you can use HTTPS data transfer.

Note: HTTPS is an advanced topic. Ensure that your HTTP upload works first and read the remote access security section in the CANedge2 Docs before proceeding. The CANedge2 Docs also describe bundled certificates and over-the-air certificate updates

Warning: We recommend that you have physical access to your CANedge2 when testing HTTPS

Warning: Depending on your WiFi/S3 setup, enabling TLS may significantly reduce your upload speed[1]

Below, we provide practical examples of how to enable HTTPS for specific S3 server types:

Enable MinIO server TLS via self-signed certificate

If you run a MinIO server, TLS is disabled by default and you'll be using a `http://` endpoint. To enable TLS on your server, you can follow the MinIO quickstart guide.

Below we use one of the examples from their guide (OpenSSL with IP address on Windows):

1. Download and extract OpenSSL
2. Create a new text file named `openssl.conf` in the folder with the `openssl.exe` file
3. Paste below into `openssl.conf`, update `IP.1` to your MinIO endpoint (excl. `http://` and port):

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no

[req_distinguished_name]
C = US
ST = VA
L = Somewhere
O = MyOrg
OU = MyOU
CN = MyServerName

[v3_req]
subjectAltName = @alt_names

[alt_names]
IP.1 = 127.0.0.1
```

4. Open the command prompt in the folder and enter the below:

```
openssl req -x509 -nodes -days 2730 -newkey rsa:2048 -keyout private.key -out public.
↪crt -config openssl.conf
```

[1] We recommend that you review your upload speed before/after adding TLS. If security is important for your use case, yet you find that your speed with TLS enabled is too low, you can consider encrypting your log files as an alternative

1. Copy the resulting `private.key` and `public.crt` files into `C:\Users\[your_user_name]\.minio\certs`
2. Rename the `public.crt` to `certs_server.p7b`[2] and copy it to the root of your device SD card
3. Update your device Configuration File to use `https://` in front of the MinIO IP endpoint

Test if the certificate is loaded in the `device.json` file and if the CANedge2 correctly uploads data. To avoid browser warnings, you can install the self-signed certificate on your PC.

Enable Cloud server TLS by downloading root certificate

For clouds (AWS, Google Cloud, ...) you can use below method to enable HTTPS for initial tests.

Warning: Note that the default root CA may change for a cloud endpoint. For production setups we strongly recommend using a custom endpoint & certificate to ensure full control

Note: For AWS S3, we provide a pre-built bundle below - but we recommend reading the full guide

How to download & deploy a certificate

1. Copy your S3 server `[endpoint]/[bucket]` into your browser. For AWS, this could e.g. be: `https://s3.us-east-1.amazonaws.com/canedge-test-bucket`
2. In Chrome, click the lock-icon next to the URL and click Certificate
3. In the popup-window, click the Certification Path
4. Here, select the top root certificate - e.g. `DigiCert Baltimore Root` in the AWS example
5. Click View Certificate, go to the Details tab and verify that it is an `RSA` type (not e.g. `ECC`)
6. Click Copy to File, then follow the guide and use the Base64-encoded format
7. Click Next, then Browse and save the `*.cer` file and rename it to `certs_server.p7b`[2]
8. Save the file to the root of your device SD
9. Update the Configuration File to use `https://` and port `443` in your server details

Important: AWS S3 certificate authority migration

AWS S3 is migrating to a new certificate authority in March 2021. To prepare, you can load both the current/upcoming certificate into a bundled certificate. To download the upcoming certificate, you can open this test link and follow the steps above. Once you've downloaded both certificates in a folder, you can install OpenSSL (as per the MinIO TLS guide) and run below command:

```
openssl crl2pkcs7 -nocrl -certfile AmazonRootCA1.cer -certfile BaltimoreCyberTrustRoot.cer -
↪out certs_server.p7b
```

You can also download our `pre-built certificate bundle for AWS S3` to enable TLS. We have tested this based on the current and upcoming certificate guidance from Amazon. However, we recommend that you go through the steps above to familiarize yourself with the process - and to ensure that nothing has changed since the build of this bundle.

[2] Before renaming the certificate, ensure that your File Explorer displays file extensions

Configuration File example

Below is an example of the CANedge2 Configuration File details for an AWS server using HTTPS:

```
"server": {
    "endpoint": "https://s3.us-east-1.amazonaws.com",
    "port": 443,
    "bucket": "amazon-bucket-name",
    "region": "us-east-1",
    "accesskey": "AKIA32WGRU62PNIX2L7T",
    "keyformat": 0,
    "secretkey": "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",
    "signed_payload": 0
  }
```

Enable Cloud server TLS via custom domain & certificate

As outlined above, if you wish to use a cloud server endpoint like e.g. AWS S3, you can use the default certificate to enable TLS. This can be OK for small scale, local setups and tests.

For production setups and large scale applications, it is recommended that you ensure full control over the certificate chain as the cloud server provider may decide to change the root CA without notice. While rare, it is a risk. To avoid this, you can use a custom domain as your end point and enable TLS by importing your preferred certificate. This is an advanced topic and we recommend involving technical staff from your cloud server provider and/or your company.

0.4.2.3 S3 server tools

The CANedge2 uploads files to an S3 server. This enables the use of any S3 compatible tool/API in managing files on the server - including below:

1. *CANcloud* - manage your S3 server log files & devices directly from your browser
2. *Mount S3 as local drive* - map your S3 server as a local drive for easier access
3. *OTA batch tool* - batch-update your Configuration Files and Firmware over-the-air
4. *S3 API* - automate e.g. log file processing and over-the-air updates via the S3 API

Note: Note that the above tools are 100% optional. You can use any other S3 compatible tools to manage your server (e.g. MinIO client, AWS CLI, ...)

CANcloud - manage devices & log files

Feature Intro | Playground

CANcloud is an open source browser tool for managing your CANedge2 devices & log files.

In this section we outline the basic functionality of CANcloud. Note that CANcloud is an optional tool for browsing your S3 data - see for example also our guide on *mounting your S3 server* as a local drive.

Logging into CANcloud

To log into CANcloud, simply go to the login page and add your S3 server details.

- If using a MinIO IP endpoint include the port (e.g. `http://192.168.0.174:9000`)
- If logging into a MinIO server without TLS, open CANcloud via this link instead (using `http://`)

Managing files

When your CANedge2 connects to your server, it'll push files with S3 object names as below:

```
[serialno]/[config-XX.YY.json]
[serialno]/[schema-XX.YY.json]
[serialno]/[device.json]
[serialno]/[session]/[split]-[epoch].MF4
```

In CANcloud, the `/` is presented as a folder structure, providing an easy overview of your data. Further, connected devices are automatically listed in the left sidebar.

- If you download a file in CANcloud, the `/` becomes a `_`
- If you upload a file in CANcloud, any `_` in the filename becomes a `/`

Example: You can upload a local file `firmware.bin` to a device folder `312AC432` in two ways:

1. Navigate to the folder `312AC432` in CANcloud and upload the file
2. Rename the file to `312AC432_firmware.bin` and upload it from the `Home` folder

Over-the-air updates

CANcloud uses the config editor tool for quick configuration of connected devices:

1. Click "Configure" next to a device in the sidebar to open the editor
2. This will auto-load the uploaded Rule Schema and Configuration File
3. You can now make edits, review changes and submit to S3 to perform an update over-the-air

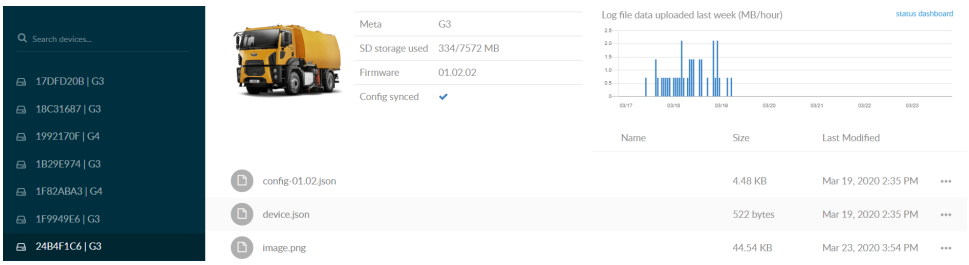
You can also perform Firmware updates over-the-air by uploading the `firmware.bin` to the device folder. For more on over-the-air updates, see the CANedge Docs and the OTA batch manager.

Note: Over-the-air updates are powerful, but require caution. Pushing e.g. the wrong WiFi/S3 details will disconnect the device until you manually reconfigure it by extracting the SD

Device meta data & image

CANcloud automatically displays meta data from your uploaded Device File. In particular, the `log_meta` field is displayed on the device page and in the sidebar (where it is searchable).

You can add a device image by uploading an image file named `image.jpg` or `image.png`.



Note: You can also hover files/folders to show info based on S3 object meta data[1]

[1] For objects, the information reflects custom S3 meta data attached to files uploaded by the CANedge2. For session folders, the information reflects a summary across the objects within. Specifically, the info includes the total number of log files, total size (as well as min/max size), the last modified S3 (time of upload to S3) and last modified SD (1st timestamp in the log file of the first and last log file in the folder).

Status dashboard

The CANcloud status dashboard helps you keep track of your connected devices:

- Check when each device last connected to your server (based on the Device File)
- Check the Firmware version of each device and whether the Configuration File is synced
- Check the free storage on your device SD cards (based on the Device File)
- View various overviews of the amount of data uploaded by your devices[2]

You can access the CANcloud status dashboard via the upper right menu.

Hosting CANcloud yourself

We host the latest version of CANcloud here, letting you easily log into your server.

However, you may want to host your own version of the tool for various reasons:

- To customize colors/logos to match your company branding
- To add custom functionality, building your own version
- To control the URL yourself for e.g. providing to end users

To host your own version, simply get the latest release and unzip it on your web server (*not* your S3 bucket). You can modify all basic CSS properties and logo files without building a new version.

For more advanced customization, you can of course also fork the source code.

Source Code | Latest Release

Mount your S3 server as a local drive

Here we explain how you can mount your S3 server as a local drive. This makes it simple to manage devices, perform OTA updates and process log files - as if your data was stored locally[1].

Table of Contents

Mount S3 on Windows (TntDrive)

TntDrive (free 30 day trial / ~60$ one-time license fee / get a discount)

TntDrive is an S3 client for mounting your S3 bucket as a Windows network drive. You can easily add any S3 compatible server - incl. AWS S3, MinIO S3 etc.

Once mapped, you can manage files as if they were stored locally - e.g. directly opening log files in data processing tools, or referencing the drive in e.g. Python/MATLAB scripts. For details on how to mount an AWS S3 or MinIO S3 server, see the video above.

[2] If you have less than 10 connected devices, the status dashboard will load the log file data by default. For more devices, you'll have to manually update the log file data via the dropdown menu (done to conserve API calls).

[1] If you're mounting an S3 cloud server, you will of course still incur costs when processing files via the mounted drive. However, if you enable caching you can save costs when processing files that have been cached due to previous use.

Other mounting tools for Windows

We generally recommend TntDrive as it's simple-to-use. However, below are various alternatives:

- S3 Drive (beta, free for personal use)
- Mountain Duck (~40$ one-time license fee)
- rclone (free, but complex to setup)

Mount S3 on Linux (s3fs)

You can also mount your S3 server on Linux. This can be done via a free open source tool, s3fs. Note, however, that the process is a bit more extensive - see our step-by-step guide.

Over-the-air batch manager

If you need to batch update your CANedge2 devices over-the-air, you can use the `canedge_manager` API/CLI. This makes it easy to update Configuration Files and/or Firmware of entire device fleets.

S3 API

You can manage your S3 server objects manually via e.g. *CANcloud* - or programmatically via the S3 API. Below we briefly outline the S3 API use cases and tools.

S3 API use cases

Typical use cases for the S3 API include:

- Performing batch over-the-air CANedge2 updates (while retaining device specific settings
- Auto-processing log files (e.g. via event triggers) - often combined with the log file APIs
- Extracting relevant parts of e.g. large CSV databases for use in plots, reports etc
- Auto-updating Configuration Files in response to certain events/patterns
- Displaying data in e.g. your browser

Tools using the S3 API

1. *CANcloud* is based on the Javascript S3 API
2. Our *OTA batch manager* lets you batch update devices over-the-air
3. Our canedge_browser lets you list log files on S3 for selected devices & time periods
4. Our API examples library on github has multiple examples of using the S3 API

0.5 Process your MDF4 data

The CANedge logs raw data in the popular MDF4 (`.MF4`) format - supported by many CAN tools.

In this section we outline some useful tools for processing your raw data:

1. *MF4 converters* - easily convert MF4 data to other formats (e.g. `csv`, `asc`, `trc`)
2. *asammdf GUI* - load, edit, DBC convert and plot your MDF4 data
3. *Python API* - automate your data processing at scale
4. Browser dashboards - visualize your data in customizable browser dashboards

You can of course also process your data in other tools, e.g. MATLAB.

0.5.1 MDF4 converters

Feature Intro

The below open source C++ executables let you easily convert raw MDF4 files into other formats.

If you're using data compression/encryption, the converters also **decompress/decrypt** your data.

Simply drag & drop files/folders (incl. nested) onto a converter - or use it in your CLI/scripts.

0.5.1.1 Download & source code

The Linux/Windows builds and source code can be found below:

Source code | README

0.5.1.2 Converter types

mdf2finalized

The CANedge records raw data as 'unsorted' and 'unfinalized'. This ensures performance and power safety. Some tools natively support unfinalized/unsorted MF4 (e.g. asammdf and our Python API), while others require that you use the `mdf2finalized` converter first (see below).

- **Vector CANalyzer** supports only finalized & sorted MF4 files, hence the mdf2finalized converter can be used to make your log files compatible (as of Vector's SP2 update)[1]
- **MATLAB's Vehicle Network Toolbox** (VNT) supports unfinalized MF4 files[2]. However, some advanced VNT use cases require that the data is already finalized & sorted (e.g. MF4 data stores). See also our MATLAB sample script

mdf2asc

Vector's ASC format is supported by various CAN tools, e.g. CANalyzer and CANape. The `mdf2asc.exe` lets you easily convert files into the ASC format for loading in such tools. With the latest SP2 for Vector's tools, you can also simply finalize the MF4 files (see above).

[1] The `mdf2finalized` tool can be used if you're using the latest SP2 for your Vector tools. If you're using older versions of Vector tools, we recommend using the `mdf2asc` converter instead as older versions of Vector's tools do not support the extended CAN ID syntax used in the CANedge MDF4 files

[2] MATLAB's Vehicle Network Toolbox supports unfinalized MF4 files as of release 2021b. In short, the tool lets you finalize/sort files as part of your script, rather than e.g. using the `mdf2finalized` converter

mdf2csv

The `mdf2csv.exe` enables quick conversion of the raw MDF4 data into a simple CSV format that you can load in text editors, Excel and other tools.

mdf2peak

The `mdf2peak.exe` lets you convert your MDF4 data to PEAK's TRC format, for use in e.g. PCAN Explorer and other PEAK tools. Default output is the `2.1` format, but you can specify `-f 1.1` via the command line.

mdf2pcap

The `mdf2pcap.exe` lets you convert raw MDF4 files into pcap format, for loading in Wireshark. Wireshark offers a range of powerful filter/analysis tools and can handle large log files seamlessly. Further, you can utilize our Wireshark plugin to e.g. convert OBD2/DBC data as in asammdf.

mdf2clx000

If you wish to convert your CANedge MDF4 log files to the CLX000 log file format, this converter lets you do that. When it runs, the converter reads an INI file in the same folder (mdf2clx000_config.ini) that follows the rules used in CLX000 configuration files. This means that you can use the settings in your old CLX000 configuration files to ensure that the output from the CANedge matches your preferred format. You can directly copy the relevant lines from the `[log]` section in your CLX000 `CONFIG.INI` file (removing any comments).

mdf2socketcan

The popular socketCAN format is supported by various open source CAN based software. With the `mdf2socketcan.exe` you can convert raw MDF4 log files for easy loading in these tools.

0.5.1.3 Encryption & compression

If you're encrypting or compressing your CANedge data, the converters will serve as a simple method for decrypting and/or decompressing the data again. All converters will natively recognize if your file is encrypted/compressed.

Note that if you have encrypted your files, you will need to add your plain form encryption password in the `passwords.json` file. You can add a single password as `default`, or e.g. add a list of device specific passwords by entering the serial number and the password as below:

```
{
    "0245BF81": "MySecret22BczPassword1234@482",
    "13FC798A": "MyOtherSecretPassword512312zZ"
}
```

0.5.1.4 CLI options

To use a converter via the CLI type the name in the command prompt to display the options.

Example: Convert a data folder (e.g. `log/`) into an output folder (e.g. `output/`) via below:

```
mdf2csv -I log -O output
```

Note: For examples on automating the converter usage, see the API examples library on github

0.5.2 asammdf GUI

Feature Intro

The asammdf GUI lets you easily load, review, DBC convert, plot and export your CANedge data.

Simply download and open the tool to use it (**no installation required**):

You can also install asammdf as below (reducing the load time for opening the GUI by 80%):

1. Install Python 3.7 for Windows (64 bit) or Linux (add to PATH)
2. Open your command prompt and write: `pip install asammdf[gui]`
3. Open your start menu, write 'asammdf' and open the GUI via the asammdf icon

0.5.2.1 Load raw data

You can directly load your raw MDF4 log file data from the CANedge in asammdf.

Loading your MDF4 data

To open a single file, click "File/Open" and browse to your `.MF4` log file.

To open multiple files (e.g. for concatenation), first click "Mode/Batch processing".

Note: If your data is compressed/encrypted (`.MFC/.MFE/.MFM`), use *mdf2finalized* to convert it to `.MF4`

Review your raw data

Once you've loaded a raw MDF4, you'll see an overview of the file in the **Channels** tab. Channel group 0 contains CAN data, while remaining channel groups contain e.g. LIN data, RTR frames etc.[1].

If you select a channel group and click the plot icon, you can display the data in a tabular form. Here you can quickly filter and analyze your data, both with relative and absolute timestamps. You can also use e.g. the CAN Trace or LIN Trace views to show multiple channel groups in one tabular display.

0.5.2.2 DBC conversion

To analyse your data, you'll need to convert it to human-readable (aka physical) form. To do so, you'll need a DBC file (CAN Database) with the decoding rules.

Converting a raw MDF4 with a DBC file

1. Select the "Bus Logging" tab, click "Load CAN database" and load your DBC
2. Click "Extract CAN signals" to save a new MDF4[1]

[1] See the CANedge Docs for details on the MDF4 log file structure and the role of each column

[1] You can optionally enable 'Ignore invalid signals', which is useful for J1939 data as it removes signals that are not actually containing valid data. For J1939/NMEA2000 data you may also consider disabling the 'Consolidated J1939' setting. By disabling this, CAN IDs that share the same PGN are no longer bundled, but are instead separated in the signal output.

The resulting MDF4 will be opened as a new tab in the GUI - ready for e.g. plotting.

J1939 & OBD2 DBC files

Usually, you'll need to be an OEM to have access to a full DBC file detailing the data parameters of a specific application, though exceptions exist:

J1939

Most heavy duty vehicles today use the standardized J1939 protocol. This means that you can typically use a J1939 DBC to decode a large share of signals across vehicle brands. We offer a *demo J1939 DBC* and a full J1939 DBC.

OBD2

Most cars let you request OBD2 PID data, which can be decoded using our free *OBD2 DBC*.

0.5.2.3 Graphical plots

Once you've converted your raw MDF4 data, you can start analysing it - e.g. via plots.

Plotting parameters

1. In the Settings tab (top menu), enable sub-plots
2. From Channels, drag & drop a parameter into the gray area to plot it
3. Optionally click the "window" icon and add more parameters to the gray area
4. Press "Shift + V" to tile the sub-plots vertically

0.5.2.4 Export

See the online documentation for details on exporting MDF4 files via asammdf.

0.5.3 Python API tools

The Python API tools let you easily automate and scale your CANedge data processing.

0.5.3.1 API modules overview

The three tools below enable most data processing use cases:

- `canedge_browser`: List log files for selected devices & time periods (from local disk or S3)
- `mdf_iter`: Extract raw CAN data from the CANedge log files (as iterable or dataframe)
- `can_decoder`: DBC-decode raw CAN data to physical values

0.5.3.2 Get started

To get started with the API tools, check out the API examples library on github.

In particular, the `data-processing/` examples show how to combine the data processing modules.