



Chipkin BACnet SC Hub CAS-1000-21

Chipkin - Enabling Integration

sales@chipkin.com

Tel: +1 866 383 1657

© 2026 CHIPKIN AUTOMATION SYSTEMS

Product version: 1.1.1

Document revision: 1.0

Issue date: 2026-Oct-15

Contents

1	What the hub does	4
1.1	Supported BIBBs	5
1.2	Services executed	5
2	Installation	6
2.1	Downloads	6
2.2	Where files go	6
2.3	The hub's certificate after installing	6
2.4	Checking the downloads	7
2.5	Supported platforms	7
2.6	Firewall	7
2.7	Running the hub	7
2.8	Starting the hub on Windows	7
2.9	Starting the hub on Linux	8
2.10	Running as a service	8
3	Quick start	9
3.1	With the Windows installer	9
3.2	From the command line	9
3.3	What you see at start-up	9
3.4	Console keys	10
4	Certificates	11
4.1	BACnet/SC support	11
4.2	The CARI file layout	11
4.3	Making the request	13
4.4	Signing with the Chipkin BACnet SC Certificate Authority	14
4.5	Installing the CA's response	14
4.6	Renewing the hub's certificate	15
4.7	Production certificates	15
4.8	Password-protected private keys	15
5	Configuration	16
5.1	Options	16
5.2	Configuration file	18
5.3	Logging	19
5.4	BACnet/SC only	19
5.5	BACnet/SC compatibility	19

6	Connecting devices	20
6.1	The set-up guide	20
6.2	Who may install a certificate	26
6.3	Setting up a device	26
6.4	The connection audit trail	27
6.5	Check that it works	27
7	Managing certificates over BACnet	28
7.1	Supported procedures	28
7.2	Replacing the hub's certificate with a new key pair	29
8	Status page	30
8.1	The health checklist	31
8.2	Update check	32
8.3	About and feedback	32
9	Security	34
9.1	BACnet/SC connections	34
9.2	Certificate revocation	34
9.3	Replacing a device and revoking its certificate	34
9.4	Web pages	34
9.5	Private keys	35
9.6	Connections to Chipkin	35
10	Troubleshooting	36
10.1	Start-up and certificates	36
10.2	Devices that can't connect	36
10.3	Web pages and console	37
10.4	Decoding BACnet/SC traffic in Wireshark	37
11	The BACnet device	39
12	Version and licensing	40
13	Contact us for support	41
A	Document history	42
B	Example configuration	43
C	Glossary of terms	44

1 What the hub does

This manual is for people who install, configure, and run the Chipkin BACnet SC Hub.

BACnet Secure Connect (BACnet/SC, ANSI/ASHRAE Standard 135-2024, Annex AB) carries BACnet over WebSockets secured with Transport Layer Security (TLS) instead of plain User Datagram Protocol (UDP). Devices on a BACnet/SC network don't talk to each other directly. Each one connects to a **hub**, which relays their traffic. That is the role a BACnet/IP broadcast domain plays for UDP devices. The Chipkin BACnet SC Hub is that hub, built on the [CAS BACnet Stack](#).

Figure 1-1 shows a BACnet/SC network. Devices and tools connect to the hub with `wss://` (TLS 1.3, mutual authentication), and the hub's own BACnet device is reached the same way.

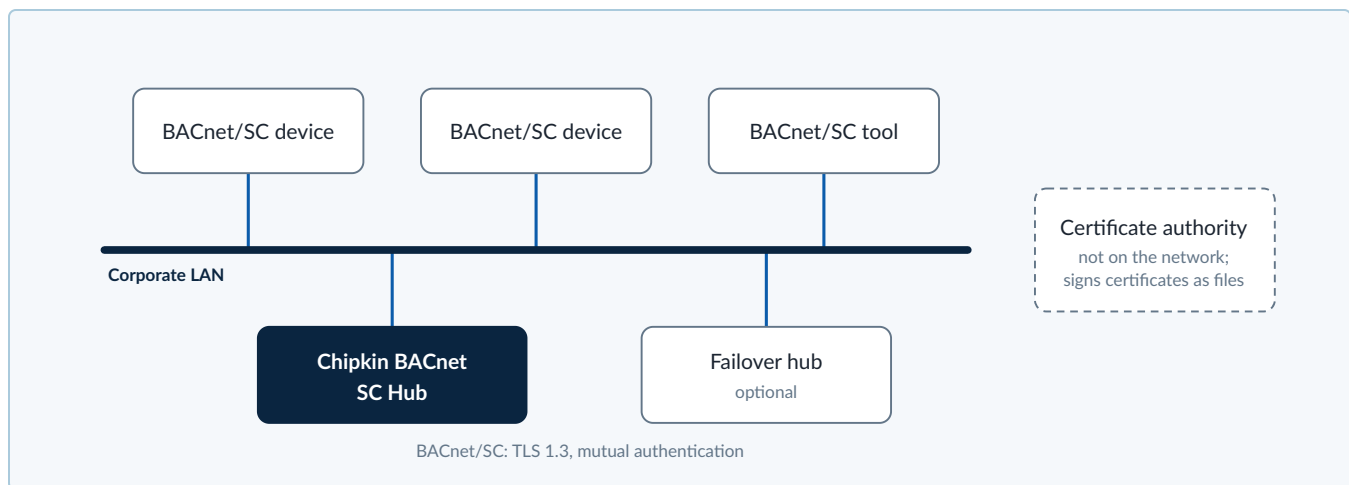


Figure 1-1. A BACnet/SC network: devices, a tool, the hub, and a second hub on the corporate LAN, with the certificate authority not connected to the network.

The hub implements the **B-SCHUB** device profile (Standard 135-2024, Annex L):

- **BACnet/SC hub function** (BACnet Interoperability Building Block, BIBB, NM-SCH-B) on Network Port 2, the hub's only data link. It listens on `wss://0.0.0.0:4443/` (TLS 1.3, mutual certificate authentication, WebSocket subprotocol `hub.bsc.bacnet.org`) and relays traffic between the connected devices, including the hub's own BACnet device.
- **Optional hub connector.** With `--sc-hub-uri`, the hub also connects out to another BACnet/SC hub (with an optional failover hub).
- **BACnet/SC only.** The hub has no BACnet/IP port. Its own BACnet device (discovery, ReadProperty, ReadPropertyMultiple, DeviceCommunicationControl, certificate management) is reached over BACnet/SC, through the hub, like every other device on the network. See section 5, "BACnet/SC only".
- **Certificates from your certificate authority (CA),** in the CARI (Certificate Authority Requirements Interchange) file format (Standard 135-2024, Annex AA.2). The hub makes its own private key and a certificate signing request (CSR), and your CA signs the request. The hub never signs certificates.
- **Certificate management over BACnet.** A client can add an issuer and replace the hub's certificate (clause 19.8.3).
- **Status page with a health checklist,** for people and for monitoring tools, and a **set-up guide** in the browser, over HTTP or HTTPS.

1.1 Supported BIBBs

BIBB	NAME
DS-RP-B	Data Sharing - ReadProperty - B
DS-RPM-B	Data Sharing - ReadPropertyMultiple - B
DM-DDB-B	Device Management - Dynamic Device Binding - B
DM-DOB-B	Device Management - Dynamic Object Binding - B
DM-DCC-B	Device Management - DeviceCommunicationControl - B
NM-SCH-B	Network Management - BACnet/SC Hub Function - B

1.2 Services executed

SERVICE	USE
ReadProperty, ReadPropertyMultiple	Read any property of any object.
Who-Is / I-Am, Who-Has / I-Have	Discovery, over BACnet/SC.
DeviceCommunicationControl	Silence or resume the device, optionally with a password.
AtomicReadFile	Read the certificate File objects.
AtomicWriteFile, WriteProperty	Certificate management only: write a certificate into File 1, 3, or 4, set the certificate File objects' <code>File_Size</code> , and write Network Port 2's <code>Command</code> (DISCARD_CHANGES, GENERATE_CSR_FILE). No other property is writable.
ReinitializeDevice	<code>ACTIVATE_CHANGES</code> or <code>WARMSTART</code> applies written certificates. Other states are refused.

The PICS (Protocol Implementation Conformance Statement) lists every object and property.

2 Installation

2.1 Downloads

Each release comes as:

DOWNLOAD	WHAT IT IS
<code>Chipkin-BACnet-SC-Hub-<version>-setup.exe</code>	The Windows installer.
<code>chipkin-bacnet-sc-hub-<version>-amd64.deb</code>	The Debian or Ubuntu package.
<code>Chipkin-BACnet-SC-Hub-<version>-linux-x64.tar.gz</code>	The installer for other Linux distributions.
<code>Chipkin-BACnet-SC-Hub.exe</code> , <code>Chipkin-BACnet-SC-Hub</code>	The bare program, to run from a folder of your choice.
<code>manual.pdf</code> , <code>fact-sheet.pdf</code> , <code>PICS.pdf</code>	This manual, the two-page fact sheet, and the PICS.
<code>sbom-windows.cdx.json</code> , <code>sbom-linux.cdx.json</code>	The software bill of materials (SBOM), in CycloneDX format.
<code>SHA256SUMS.txt</code>	SHA-256 checksums of every download.

To install:

- **Windows:** run the installer. It adds Start-menu shortcuts that run the hub in a console window and open this manual, and optionally a firewall rule for TCP 4443.
- **Debian or Ubuntu:** run `sudo apt install ./chipkin-bacnet-sc-hub-<version>-amd64.deb`. It creates the `bacnethub` user the hub runs as.
- **Other Linux distributions:** unpack the `.tar.gz` and run `sudo ./install.sh`, which installs the same files as the package. `sudo ./uninstall.sh` removes the hub (`--purge` also removes settings and certificates).

The SBOM lists the exact CAS BACnet Stack, OpenSSL, libwebsockets, and other library versions.

2.2 Where files go

ITEM	WINDOWS	LINUX
Program	<code>C:\Program Files\Chipkin\BACnet SC Hub</code>	<code>/opt/chipkin-bacnet-sc-hub</code>
Settings	<code>C:\ProgramData\Chipkin\BACnetSCHub\hub.conf</code>	<code>/etc/chipkin-bacnet-sc-hub/hub.conf</code>
Certificates	<code>C:\ProgramData\Chipkin\BACnetSCHub\certs</code>	<code>/etc/chipkin-bacnet-sc-hub/certs</code>
Logs	<code>C:\ProgramData\Chipkin\BACnetSCHub\logs\hub.log</code>	<code>/var/log/chipkin-bacnet-sc-hub/hub.log</code>

Settings and certificates are kept when you uninstall or reinstall.

2.3 The hub's certificate after installing

When the certificate folder is empty, both installers run `--generate-csr`: the hub makes its private key and a CSR, and writes the request, `hub-csr-request.zip`, in the certificate folder.

**IMPORTANT**

BACnet/SC can't start until the CA's response to that request is installed, so until then the hub is unreachable over BACnet. The status page shows what to fix. See section 4, "Certificates".

2.4 Checking the downloads

The Windows program and the Windows installer are code-signed by Chipkin (Azure Artifact Signing, SHA-256 with a timestamp). Check them with `Get-AuthenticodeSignature <file>`. To check any download against its checksum, run `sha256sum -c SHA256SUMS.txt --ignore-missing`.

2.5 Supported platforms

Windows 11 or later (x64). Linux x64: the release build is made on the current Ubuntu long-term support (LTS) release, so it needs a distribution with that glibc version or newer. For another platform, contact Chipkin.

2.6 Firewall

Allow in:

PORT	PROTOCOL	FOR
4443	TCP	BACnet/SC devices connecting to the hub (<code>--sc-port</code>).
8080	TCP	The status page and the hub's other web pages (<code>--http-port</code>). Only if you bind it beyond 127.0.0.1.

The Windows installer can add the first rule for you. The hub opens no UDP port.

**CAUTION**

Open TCP 8080 only after reading section 9, "Security". The hub's web pages have no authentication, and without `--http-tls` they are served over plain HTTP.

2.7 Running the hub

Started from a console window, the hub opens the status page in your default web browser once its web server is up (on Linux, only in a desktop session). The address is printed either way. A hub started from a script, a pipe, or a service never opens a browser.

Leave the window open while the hub runs. Press `q` (or Ctrl+C) to stop it.

2.8 Starting the hub on Windows

Start the Chipkin BACnet SC Hub shortcut from the Start menu. It runs the hub with the installed settings, the same as:

```
cd /d C:\ProgramData\Chipkin\BACnetSCHub
"C:\Program Files\Chipkin\BACnet SC Hub\Chipkin-BACnet-SC-Hub.exe" --config hub.conf
```

Relative paths in `hub.conf` (`sc-cert-dir = certs`, `log-file = logs\hub.log`) are relative to that folder. Only SYSTEM, Administrators, and the user who installed the hub can open the folder, so another user needs an Administrator command prompt.

2.9 Starting the hub on Linux

After installing the `.deb` or running `install.sh`, start the hub as the `bacnethub` user, which owns its settings, certificates, and log:

```
sudo -u bacnethub /opt/chipkin-bacnet-sc-hub/Chipkin-BACnet-SC-Hub --config /etc/chipkin-bacnet-sc-hub/hub.conf
```

Its output also goes to `/var/log/chipkin-bacnet-sc-hub/hub.log`.

2.10 Running as a service

On Windows the hub can run as the `BACnetSCHub` service, which starts on boot and restarts on failure. To set it up:

1. Open an Administrator command prompt.
2. Run `"C:\Program Files\Chipkin\BACnet SC Hub\Chipkin-BACnet-SC-Hub.exe" --install-service --config C:\ProgramData\Chipkin\BACnetSCHub\hub.conf`.
3. Start the `BACnetSCHub` service, or restart the computer.

The service runs with the configuration file's folder as its working directory, so relative paths in the configuration file are relative to that folder. A service has no console, so set `log-file` to keep a log, and if the hub's key is password-protected, set `sc-key-password` (see section 4, "Password-protected private keys").

To remove the service, run `Chipkin-BACnet-SC-Hub --uninstall-service` from an Administrator command prompt. Uninstalling the hub also removes it.

The Linux installers set up no service. SIGTERM and SIGINT stop the hub cleanly, the same as pressing `q`.

3 Quick start

Both paths below end with the hub listening and devices connected. "Set-up guide step N" means one of the four numbered steps on the set-up guide's page (see section 6, "The set-up guide").

3.1 With the Windows installer

1. Run `Chipkin-BACnet-SC-Hub-<version>-setup.exe`. The installer makes the hub's private key and its request.
2. Start the Chipkin BACnet SC Hub shortcut from the Start menu. The status page opens in your web browser.
3. Open the set-up guide: click **Set-up** in the app bar, or go to <http://127.0.0.1:8080/setup>.
4. In set-up guide step 2, download `hub-cari-request.zip`.
5. Have your CA sign the request (see section 4, "Signing with the Chipkin BACnet SC Certificate Authority"). The CA gives you a response zip.
6. In set-up guide step 3, upload the CA's response.
7. On the status page, open the health checklist and check that every item passes.
8. Give each device its own certificate from the same CA, the issuer certificate, and the hub URI (set-up guide step 4 lists them; see section 6, "Connecting devices").
9. Confirm that the devices appear under **Connected devices** on the status page.

3.2 From the command line

Work from the folder where you unpacked the program. On Windows the program is `Chipkin-BACnet-SC-Hub.exe`. To start an installed hub instead, see section 2, "Running the hub".

1. Make the hub's private key and its request: `Chipkin-BACnet-SC-Hub --generate-csr`
2. Send `certs/hub-cari-request.zip` to your CA.
3. Install the CARI response the CA sends back: `Chipkin-BACnet-SC-Hub --import-cari hub-cari-request-response.zip`
4. Start the hub: `Chipkin-BACnet-SC-Hub`
5. Check the health checklist on the status page, at <http://127.0.0.1:8080/>.
6. Connect the devices (see section 6, "Connecting devices").

The set-up guide, at <http://127.0.0.1:8080/setup>, can replace steps 2 and 3. It lets you download the request and upload the CA's response, lists what each device needs, and shows why a connection attempt failed.

3.3 What you see at start-up

A typical start-up:

```
Chipkin BACnet SC Hub v1.1.1
CAS BACnet Stack version: 6.0.23.0
Common helper (common/) version: 3.1.0
2026-10-07 18:10:21 [OK] Device 389022 ("Chipkin BACnet SC Hub") ready. Vendor ID 389. Press 'h' for
help, 'm' for a health/metrics snapshot.
2026-10-07 18:10:21 [OK] status page: http://127.0.0.1:8080/ - health checklist, connected devices
and traffic
2026-10-07 18:10:21 [INFO] opened the status page in your web browser
2026-10-07 18:10:21 [OK] BACnet/SC: listening for WebSocket/TLS connections on wss://0.0.0.0:4443/
(subprotocol "hub.bsc.bacnet.org", TLS 1.3, mutual auth)
```

On a terminal the lines are coloured by level: errors red, warnings yellow, `[OK]` lines green, information in the terminal's own colour. The colour is left out when the output goes to a file or a pipe, when the `NO_COLOR` environment variable is set (to anything), and always in the `--log-file` copy.

At start-up the hub also checks its certificates and logs each one's subject, days until expiry (a warning under 30 days), whether the private key matches, and the subjectAltName (subject alternative name) entries.

3.4 Console keys

KEY	ACTION
<code>h</code>	Help and version.
<code>m</code>	Health and metrics snapshot.
up, down	Change Analog Input 1 by 1.1.
<code>q</code>	Quit.

4 Certificates

4.1 BACnet/SC support

BACnet/SC runs over TLS with **mutual authentication**: the hub and every device present a certificate, and each side accepts the other only if that certificate was signed by an issuer (a CA) it trusts. Without its own certificate, the hub can't listen for BACnet/SC, so nothing can reach it over BACnet. Its status page and set-up guide still work, and the status page shows what to fix.

The hub never signs certificates. A CA does. You can use:

- the **Chipkin BACnet SC Certificate Authority** (part number CAS-1000-22), Chipkin's CA application for BACnet/SC sites;
- **any CARI-compatible certificate authority**, including your own public key infrastructure (PKI). See the Production certificates guide.

Getting the hub its certificate takes four steps:

1. **Make the request.** Run `Chipkin-BACnet-SC-Hub --generate-csr`. It makes the hub's private key and a CSR, and writes `hub-cari-request.zip` in the certificate folder. The installers do this for you. The zip holds the CSR only; the private key never leaves the hub.
2. **Send it to your CA.** Download `hub-cari-request.zip` from the set-up guide (set-up guide step 2) or take it from the certificate folder, and send it to the CA.
3. **Install the CA's response.** The CA sends back a CARI response zip. Install it with `--import-cari <response.zip>` and restart the hub, or upload it in the set-up guide (set-up guide step 3), which installs it and starts using it straight away.
4. **Check.** Confirm that the status page says the hub is listening.

Both ways of installing check the response first: the certificate in it must be for **this hub's request** (its public key matches the hub's CSR), it must not have expired, and one of the issuer certificates in the zip must have signed it. Anything else is refused and nothing is written.

The hub's certificate can also be installed over BACnet (see section 7, "Managing certificates over BACnet").

4.2 The CARI file layout

Certificate files use the **CARI** format (Standard 135-2024, Annex AA.2, added to Standard 135-2020 by Addendum cs). CARI is the folder layout BACnet uses to take CSRs to a CA and bring the certificates back. Its root folder is `cert1/`:

- **Folders:** one `device-<instance>/` folder per device, and one `port-<id>/` folder per BACnet/SC port in it. A `hub/` folder in a port folder marks a hub function, and a `router/` folder in a device folder marks a device that routes between BACnet/SC networks. Both are empty.
- **Files in a port folder:** `csr-<name>.pem` (the request), `opr-<name>.pem` (the operational certificate the CA returns) and, optionally, `key-<name>.pem` (the private key). The files are in Privacy-Enhanced Mail (PEM) format.
- **The issuer:** `cert1/issuer/iss-1.pem` (and `iss-2.pem`).

A **CARI request** is a zip of such a tree with the CSRs in it. The CA's **CARI response** is the same tree with the `opr-` certificates and the issuer certificate(s) added.

The hub's request, `hub-cari-request.zip`, holds the CSR and never the key:

```
hub-cari-request.zip
├─ cert1/
│  └─ device-389022/
│     └─ port-2/
│        └─ hub/
│           └─ csr-hub.pem
└─ request-notes.txt
```

empty: marks a hub function
the hub's certificate signing request
a note to the CA

The CA's response is the same tree with the hub's certificate and the CA's own certificate added:

```
hub-cari-request-response.zip
├─ cert1/
│  └─ device-389022/
│     └─ port-2/
│        └─ hub/
│           └─ csr-hub.pem
│           └─ opr-hub.pem
└─ issuer/
   └─ iss-1.pem
└─ request-notes.txt
└─ response-notes.txt
```

added: the hub's certificate
added: the CA's certificate
added: a note from the CA

The hub's certificate folder (`--sc-cert-dir`, default `./certs`) is a CARI tree for the hub itself, with the hub's own working files beside it:

```
certs/
├─ cert1/
│  └─ device-389022/
│     └─ port-2/
│        └─ hub/
│           └─ csr-hub.pem
│           └─ key-hub.pem
│           └─ opr-hub.pem
└─ issuer/
   └─ iss-1.pem
   └─ iss-2.pem
└─ hub-cari-request.zip
└─ issuer-crl.pem
└─ key-hub-pending.pem
└─ readme.txt
└─ trusted-issuers.pem
```

only CARI files in here
empty: Network Port 2 is a hub function
the hub's certificate signing request
the hub's private key (private)
the hub's certificate
the CA that signed the hub's certificate
the second issuer slot
the request to send to the CA
a certificate revocation list (optional)
a new key waiting for its certificate (private)
a walkthrough of the folder
every issuer the hub trusts, written by the hub

The table says what each file is:

FILE	WHAT IT IS
<code>cert1/device- <instance>/port-2/hub/</code>	Empty: Network Port 2 is a hub function.

FILE	WHAT IT IS
<code>cert1/device-<instance>/port-2/key-hub.pem</code>	The hub's private key, made by <code>--generate-csr</code> . Private.
<code>cert1/device-<instance>/port-2/csr-hub.pem</code>	The hub's CSR (File 2, <code>Certificate_Signing_Request_File</code>).
<code>cert1/device-<instance>/port-2/opr-hub.pem</code>	The hub's certificate, from the CA's response (File 1, <code>Operational_Certificate_File</code>).
<code>cert1/issuer/iss-1.pem</code>	The CA that signed it (File 3, <code>Issuer_Certificate_Files</code>). Devices need a copy to trust the hub.
<code>cert1/issuer/iss-2.pem</code>	File 4: the CA response's second issuer. With one issuer it is a copy of <code>iss-1.pem</code> , so no old issuer stays trusted. It can also be changed over BACnet.
<code>hub-cari-request.zip</code>	The hub's CARI request, to send to the CA. Holds the CSR, never the key.
<code>key-hub-pending.pem</code>	Only after a <code>GENERATE_CSR_FILE</code> over BACnet: the new key behind <code>csr-hub.pem</code> , until a certificate for it is installed. Private.
<code>trusted-issuers.pem</code>	Written by the hub: every issuer from both slots, which is what TLS trusts.
<code>issuer-crl.pem</code>	Optional, from your CA: revoked device certificates, as a certificate revocation list (CRL). See section 9, "Certificate revocation".
<code>readme.txt</code>	A walkthrough of the folder, including which files are private.



WARNING

`key-hub.pem` and `key-hub-pending.pem` are the hub's private keys. Keep them on the hub and never send them to anyone, including your CA. The CA needs only `hub-cari-request.zip`, which holds the CSR and never the key.

`<instance>` is the hub's device instance (`--deviceID`, default 389022). If you change the device instance afterwards, the hub keeps using the existing folder and logs a note. Rename `device-<old>` to `device-<new>` to match.

4.3 Making the request

```
Chipkin-BACnet-SC-Hub --generate-csr
Chipkin-BACnet-SC-Hub --generate-csr --cert-hub-uri wss://bacnet-hub.example.site:4443/
```

The request is for an Elliptic Curve Digital Signature Algorithm (ECDSA) P-256 key, signed with SHA-256. Its subject Common Name is the device name (`--device-name`). It asks for the TLS purposes a hub needs (serverAuth and clientAuth) and for a subjectAltName that holds `localhost`, `127.0.0.1`, this computer's host name, and the host of the hub URI devices connect to. That host comes from `--cert-hub-uri`, or by default this computer's IPv4 address. The CA decides what the certificate finally contains.

Running `--generate-csr` again keeps the key and writes the zip again from the same request; it never replaces a key. To make a new request (for another host name, for example), move `key-hub.pem` and `csr-hub.pem` out of the folder first.



CAUTION

Move `key-hub.pem` only when you intend to replace the hub's certificate. The current certificate stops working without it.

For an installed hub, run the commands with its settings, as the account that owns the certificate folder:

```
# Linux
sudo -u bacnethub /opt/chipkin-bacnet-sc-hub/Chipkin-BACnet-SC-Hub \
  --config /etc/chipkin-bacnet-sc-hub/hub.conf --generate-csr
```

```
REM Windows, as the user who installed the hub (or from an Administrator command prompt)
cd /d C:\ProgramData\Chipkin\BACnetSCHub
"C:\Program Files\Chipkin\BACnet SC Hub\Chipkin-BACnet-SC-Hub.exe" --config hub.conf --generate-csr
```

4.4 Signing with the Chipkin BACnet SC Certificate Authority

The Chipkin BACnet SC Certificate Authority takes `hub-cari-request.zip` in and gives `hub-cari-request-response.zip` back. Do this for each hub (a primary and a failover hub each have their own key and certificate). The Certificate Authority's user guide has the detail.

1. Copy `hub-cari-request.zip` to the computer with the Certificate Authority.
2. Open your CA, click **Sign a request**, and choose `hub-cari-request.zip`.
3. Check the table: the request shows as a hub, with its host names or addresses.
4. Choose how long the certificate is valid, then click **Sign 1 and save the response**.
5. Save the CA file again (**Save**, or Ctrl+S).
6. Copy `hub-cari-request-response.zip` back to the hub's computer and install it (see "Installing the CA's response" in this section).

4.5 Installing the CA's response

On the command line:

```
Chipkin-BACnet-SC-Hub --import-cari hub-cari-request-response.zip
```

It prints the certificate's subject, issuer, expiry, and SHA-256 fingerprint, and writes `opr-hub.pem`, `cert1/issuer/iss-1.pem`, and `iss-2.pem` (the response's second issuer, or a copy of the first). Restart the hub to use it. For an installed hub, run it as in "Making the request", with `--import-cari <file>` in place of `--generate-csr`. On Linux, put the zip somewhere the `bacnethub` user can read, such as `/tmp`.

In the **set-up guide**: set-up guide step 3, *Install the CA's response*, takes the zip and installs it while the hub runs. BACnet/SC reloads with the new certificate, and connected devices reconnect. See section 6, "Who may install a certificate".

**CAUTION**

The issuers in the response replace the hub's issuer slots. Devices whose certificates come from another CA are then no longer accepted. To keep trusting a second CA (while a site moves to a new one, for example), use a response with both issuers in it if your CA can include both issuers; otherwise add the second issuer over BACnet (see section 7, "Managing certificates over BACnet").

4.6 Renewing the hub's certificate

The health checklist's **Certificates not expiring soon** item becomes a warning when the hub's certificate or an issuer's certificate has 30 days or less left, and a failure once one has expired. The start-up log gives the same warning.

- **Same key.** Have your CA sign the hub's request again and install the response (see "Installing the CA's response" in this section). With the Chipkin BACnet SC Certificate Authority, **Renew** in **Issued certificates** uses the same request and the same key.
- **New key, from the command line.** Move `key-hub.pem` and `csr-hub.pem` out of the folder, run `--generate-csr`, have your CA sign the new request, and install the response (see "Making the request" in this section).
- **New key, while the hub keeps running.** Use `GENERATE_CSR_FILE` over BACnet (see section 7, "Replacing the hub's certificate with a new key pair").

4.7 Production certificates

The Production certificates guide covers running the hub with your own CA in more depth: choosing a CA, root and intermediate layouts, signing the hub's request with your own PKI, rotating certificates and changing the CA without an outage, revocation, what the certificates should contain, and protecting private keys.

4.8 Password-protected private keys

A private key may be stored encrypted (the file starts `-----BEGIN ENCRYPTED PRIVATE KEY-----`), for example when you replace the hub's key with one protected by your own tools. The hub asks for its password **once**, when it starts, with the typing hidden, and checks it by decrypting the key. A wrong password is asked for again, up to three times. The hub keeps the password in memory and uses it everywhere the key is needed (the listener, the connector's every connect attempt, the HTTPS status page), so it is never asked for again during the run.

A service has no console to ask on: set the password in the configuration file as `sc-key-password` (configuration file only, like `dcc-password`; restrict the file's permissions as described in section 5, "Configuration file"). When it is set, the hub doesn't ask.

The same password unlocks the set-up guide's certificate installation from any computer that reaches it over HTTPS. Without one, it works only on the hub's own computer.

5 Configuration

Chipkin-BACnet-SC-Hub [options]

5.1 Options

Device

OPTION	FUNCTION	LEGAL VALUES
<code>--deviceID <n></code>	The BACnet device instance.	0 to 4194302, 389022
<code>--device-name <name></code>	The Device's <code>Object_Name</code> .	1 to 128 bytes of UTF-8, Chipkin BACnet SC Hub
<code>--ip-address <address or name></code>	The network adapter the hub advertises.	An IPv4 address of this computer or an adapter name, the first IPv4 adapter
<code>--config <path></code>	Read settings from a configuration file.	A path, none
<code>--no-update-check</code>	Don't ask Chipkin's update service for a newer version.	None (a switch), the hub checks

* Bolded values are defaults.

The device name must be unique on the BACnet internetwork, so name each hub. It is 1 to 128 bytes of UTF-8; for a non-ASCII name use `device-name` in a configuration file saved as UTF-8.

`--ip-address` matters on a computer with several networks. Give an IPv4 address of this computer (`192.168.1.20`) or an adapter name (`Ethernet` , `eth0`). It sets the hub URI that `--generate-csr` asks for in the certificate's subjectAltName and that the set-up guide gives devices. The listeners still accept on every adapter. A name that matches nothing stops the hub and lists the adapters.

For `--config`, see "Configuration file" in this section. For `--no-update-check`, see section 8, "Update check".

BACnet/SC

OPTION	FUNCTION	LEGAL VALUES
<code>--sc-port <n></code>	The BACnet/SC hub (<code>wss://</code>) port.	A TCP port number, 1 to 65535, 4443
<code>--sc-cert-dir <dir></code>	The certificate folder.	A folder path, ./certs
<code>--sc-network-number <n></code>	The <code>Network_Number</code> of Network Port 2.	1 to 65534, not set
<code>--sc-hub-uri <wss://host:port/></code>	Also connect out to this BACnet/SC hub.	A <code>wss://</code> URI, off
<code>--sc-failover-uri <wss://host:port/></code>	The failover hub for <code>--sc-hub-uri</code> .	A <code>wss://</code> URI, off
<code>--sc-max-hub-connections <n></code>	The maximum number of BACnet/SC devices connected at once.	A number of devices, 4
<code>--sc-rate-limit <n></code>	The maximum new connection attempts per second from any one source address.	0 to 65535 (0 = no limit), 10

OPTION	FUNCTION	LEGAL VALUES
<code>--sc-rate-limit-total <n></code>	The maximum new connection attempts per second for the whole listener.	0 to 65535 (0 = no limit), 50
<code>--sc-accept-hub-without-hello</code>	Compatibility: let <code>--sc-hub-uri</code> connect to a hub that omits the Hello option.	on or off
<code>--sc-accept-device-without-hello</code>	Compatibility: accept a device whose Connect-Request omits the Hello option.	on or off
<code>--sc-keylog-file <file></code>	Debugging only: write each BACnet/SC connection's TLS session secrets to a file.	A file path, off

A network number set with `--sc-network-number` is reported with quality `configured`. Unset, Network Port 2 reports 0, quality `unknown`.

Connection attempts over a rate limit are refused before the TLS handshake.

For the two compatibility options, see "BACnet/SC compatibility" in this section. `--sc-keylog-file` lets Wireshark decrypt a capture and can be given on the command line only; see section 10, "Decoding BACnet/SC traffic in Wireshark".

Web pages

OPTION	FUNCTION	LEGAL VALUES
<code>--http-port <n></code>	The port for the status page and the hub's other web pages.	A TCP port number, 1 to 65535, 8080
<code>--http-bind <addr></code>	The interface the web pages are served on.	An IP address of this computer, 127.0.0.1
<code>--http-tls</code>	Serve the web pages over HTTPS.	on or off
<code>--http-tls-cert <file></code> , <code>--http-tls-key <file></code>	The certificate and key for <code>--http-tls</code> .	File paths, the hub's certificate and key

Read section 9, "Web pages", before changing `--http-bind`.

Logging

OPTION	FUNCTION	LEGAL VALUES
<code>--log-file <path></code>	Also write everything the hub prints to this file.	A file path, none
<code>--log-max-size-mb <n></code>	Rotate the log file at this size.	1 to 4096 (MB), 10
<code>--log-max-files <n></code>	Keep this many rotated log files (<code><path>.1</code> , <code>.2</code> , ...).	0 to 100, 5
<code>--xml</code>	Print every received and sent BACnet message as an indented XML block instead of a one-line summary (very verbose).	None (a switch), off

See "Logging" in this section.

Commands that run and exit

OPTION	FUNCTION	LEGAL VALUES
<code>--generate-csr</code>	Make the hub's private key, its CSR, and <code>hub-cari-request.zip</code> for the CA.	None
<code>--cert-hub-uri</code> <code><wss://host:port/></code>	The hub URI devices connect to, for <code>--generate-csr</code> .	A <code>wss://</code> URI, this computer's IPv4 address and <code>--sc-port</code>
<code>--import-cari <zip></code>	Install the CA's CARl response after checking it.	The path of a CARl response zip
<code>--inspect <file></code>	Say what a certificate, CSR, key, <code>.pfx</code> , or CARl zip is, and check it against this hub.	A file path
<code>--install-service</code>	Windows: set up the <code>BACnetSCHub</code> service.	None
<code>--uninstall-service</code>	Windows: stop and remove the service.	None
<code>--help</code> , <code>-h</code>	Show the help and exit.	None
<code>--version</code>	Show version information and exit.	None

With a key already there, `--generate-csr` writes the zip from the existing request, and asks for the host of `--cert-hub-uri` in the certificate's subjectAltName (see section 4, "Making the request"). Restart the hub after `--import-cari`. `--inspect` is the set-up guide's file inspector: it lists every field and exits with 1 if a check fails. For the service commands, see section 2, "Running as a service".

An unknown option, a stray argument, or an option missing its value is an error. The switches `--http-tls`, `--sc-accept-hub-without-hello`, and `--sc-accept-device-without-hello` mean "on" alone and also take `on` or `off`, so the command line can turn off what the configuration file turned on.

5.2 Configuration file

`--config <path>` reads `key = value` lines. A line starting with `#` is a comment; a `#` later in a line is part of the value, so passwords may contain one.

Each option in the Device, BACnet/SC, Web pages, and Logging tables has a configuration key of the same name without the leading `--`. The exceptions are:

OPTION OR KEY	DIFFERENCE
<code>--deviceID</code>	The key is <code>device-id</code> .
<code>--no-update-check</code>	The key is <code>check-for-updates = false</code> .
<code>--config</code> , <code>--sc-keylog-file</code> , <code>--xml</code>	Command line only; there is no key.
<code>dcc-password</code> , <code>sc-key-password</code>	Configuration file only; there is no option.

`example.conf` (shipped with the program) lists the keys with their defaults, and Appendix B, "Example configuration", shows a complete file. Keys that are on or off take `true` or `false` (or `on` or `off`, `yes` or `no`, `1` or `0`). A command-line option always wins over the configuration file. An unknown key or bad value is logged and skipped.

`dcc-password` and `sc-key-password` can only be set in the configuration file, so they never show up in process listings or shell history. `sc-key-password` is the password of an encrypted private key (see section 4, "Password-protected private keys"). `dcc-password` protects DeviceCommunicationControl and ReinitializeDevice; it is 1 to 20 characters.

Restrict the file's permissions when it holds a password: `chmod 600 hub.conf`, or on Windows `icacls hub.conf /inheritance:r /grant:r *S-1-5-18:F *S-1-5-32-544:F "%USERNAME%:F"`, which keeps access for SYSTEM, Administrators, and you. The hub warns at start-up if the file looks readable by other users. The Windows installer already restricts `C:\ProgramData\Chipkin\BACnetSCHub` to SYSTEM, Administrators, and the user who installed the hub.

5.3 Logging

Everything the hub prints goes to the console. With `--log-file <path>` (configuration key `log-file`) it also goes to that file, which is appended to across restarts. When the file reaches `--log-max-size-mb` (default 10 MB) it is renamed `<path>.1` (the previous `.1` becomes `.2`, and so on) and a new file is started; `--log-max-files` (default 5) old files are kept. Every line written through the hub's logger starts with a UTC timestamp and a level (`[INFO]`, `[OK]`, `[WARNING]`, `[ERROR]`). Connection audit lines start with `SC audit:`. The log file never contains colour codes, even when the console copy is coloured.

5.4 BACnet/SC only

The hub's only data link is BACnet/SC. Its only Network Port is Network Port 2 "BACnet SC", and its own BACnet device is reachable only over BACnet/SC: discovery (Who-Is and I-Am) and every other service go through the hub, like every device on the network. A BACnet tool reaches it by connecting to the hub as a BACnet/SC device (it needs a certificate from your CA, like any device). The hub doesn't send an I-Am at start-up, since no device is connected yet; devices find it with Who-Is once they connect.

The hub warns a few seconds after start-up if BACnet/SC isn't listening (no certificate yet, for example), because the device can't then be reached over BACnet at all. BACnet/SC keeps retrying every 5 seconds, and the status page shows what to fix.

5.5 BACnet/SC compatibility

Standard 135-2024 (clause AB.2.2) requires the BACnet/SC Connect-Request and Connect-Accept messages to carry a *Hello* option. By default the hub follows the standard strictly and refuses a connection that leaves it out.

`--sc-accept-hub-without-hello` (configuration key `sc-accept-hub-without-hello = true`) relaxes this for the **hub connector** only: when the hub connects out to another hub (`--sc-hub-uri`), it accepts that hub's Connect-Accept without Hello and treats the other hub as having no optional capabilities. Turn it on only for a hub that needs it; the hub logs a warning at start-up while it is on.

`--sc-accept-device-without-hello` (configuration key `sc-accept-device-without-hello = true`) does the same for **devices** connecting to this hub: it accepts a Connect-Request without Hello and records the device as having no optional capabilities. YABE (Yet Another BACnet Explorer, a third-party BACnet tool) sends such a request. Without the option, such a device is refused and the hub logs `Connect-Request refused by the hub: "Connect messages require the Hello destination option"`. Turn it on only while a device like that needs to connect; the hub logs a warning at start-up while it is on.

The two settings are independent; either, both, or neither can be on.

6 Connecting devices

Each BACnet/SC device needs three things:

1. **Its own certificate and private key**, from the **same CA** as the hub's certificate (or from a CA in the hub's other issuer slot).
2. **The issuer certificate**, so the device trusts the hub: `iss-1.pem` (and `iss-2.pem`, if the hub has two issuers). The CA's response for the device usually includes it; the set-up guide also offers it for download.
3. **The hub URI**, `wss://<hub address>:4443/`.

There are two ways for a device to get its certificate and key:

- **The device makes its own key.** The device makes its key and a CSR, or a CARI request. The CA signs the request and sends back the certificate (`opr-<name>.pem` in a CARI response). The key stays on the device.
- **The CA makes the key and the certificate.** For a device or tool that can't make a request, the CA makes both and gives you a file that holds the private key. Carry that file privately, install it on the device, then delete the copies. The Certificate Authority's user guide describes this under "Devices that can't make a request".



IMPORTANT

Give each device its own certificate. The hub tells devices apart by their certificates.

The hub trusts every certificate its issuers signed, straight away; there is nothing to add on the hub for a new device. Revoked certificates are refused when you use a revocation list (see section 9, "Certificate revocation").

6.1 The set-up guide

On the hub's computer, open <http://127.0.0.1:8080/setup> (the status page links to it, and every page's app bar calls it **Set-up**). It takes you through four steps.

Step 1: This hub's certificate. This step explains how trust works and shows the hub's certificate and its issuer's certificate, with every field and every check (Figure 6-1).

CHIPKIN

BACnet SC Hub

Status

Set-up

About

CERTIFICATES

Chipkin BACnet SC Hub set-up guide

Chipkin BACnet SC Hub · v1.1.1 · device 389022

1 This hub's certificate

2 Get it signed

3 Install the CA's response

4 Connect devices

File inspector

Why did a connection fail?

Support report

This hub's certificate

BACnet/SC connections use TLS 1.3 with mutual authentication: the hub and each device show a certificate, and each accepts the other only if a Certificate Authority (CA) it trusts signed it. This hub does not sign certificates. Your CA signs the hub's certificate and every device's.

✓ The hub has a certificate, and every check passes.

Subject

CN=Main Street hub,O=Example Building Controls

Issued by

CN=Main Street site CA,O=Example Building Controls

Valid until

2029-01-14 17:45:38 UTC (824 days left)

Certificate Authority
signs both (iss-1.pem)

Your device
its certificate + key

This hub
opr-hub.pem + its key

TLS 1.3, both directions

Device

Chipkin BACnet SC Hub (instance 389022)

Software

Chipkin BACnet SC Hub v1.1.1, CAS BACnet Stack 6.0.23.0

BACnet/SC port

4443

Certificate folder

C:/ProgramData/Chipkin/BACnetSCHub/certs (CARI (cert1/))

Figure 6-1. Set-up guide step 1, This hub's certificate: how trust works, and the hub's certificate and its issuer's certificate with their checks.

Step 2: Get it signed. In Figure 6-2 you download `hub-cari-request.zip` (the hub's CSR, never its key) to send to your CA. If the hub has no request yet, the page says to run `--generate-csr`.

CAS-1000-21-Chipkin-BACnet-SC-Hub-Manual-2026-10-15 · Rev 1.0

Page 21 of 44

 BACnet SC Hub

StatusSet-upAbout

CERTIFICATES

Chipkin BACnet SC Hub set-up guide

Chipkin BACnet SC Hub · v1.1.1 · device 389022

1 This hub's certificate2 **Get it signed**3 Install the CA's response4 Connect devicesFile inspectorWhy did a connection fail?Support report

Get the hub's certificate signed

Do this if the hub has no certificate yet, or to renew it. The hub's request holds its certificate signing request (CSR) only. Its private key never leaves the hub.

✓

The hub has a certificate request (cert1/device-389022/port-2/csr-hub.pem).
To renew, download it again and send it to your CA.

1

Download the hub's request:

Download hub-cari-request.zip

2

Send it to your Certificate Authority: the **Chipkin BACnet SC Certificate Authority, BACCARI** (BACnet International), or any CA that reads CARI files.

3

The CA sends back a CARI response zip. Install it in step 3.

The files use the CARI format (ANSI/ASHRAE 135-2024 Annex AA.2): csr- request, opr- certificate, iss-1 issuer.

Next: install the CA's response

Chipkin BACnet SC Hub 1.1.1 Chipkin Automation Systems Support: support@chipkin.com

Figure 6-2. Set-up guide step 2, Get it signed: the download of hub-cari-request.zip to send to your CA.

Step 3: Install the CA's response. In Figure 6-3 you upload the CARI response zip. The hub checks that it is for this hub's request and that the CA signed it, then installs it and starts using it. Connected devices reconnect.

CERTIFICATES

Chipkin BACnet SC Hub set-up guide

Chipkin BACnet SC Hub · v1.1.1 · device 389022

1 This hub's certificate 2 Get it signed **3 Install the CA's response** 4 Connect devices File inspector Why did a connection fail? Support report

Install the CA's response

Upload the CARL response zip your CA sent back. The hub checks that it answers this hub's request and that the CA signed it, then installs it and starts using it. Connected devices reconnect.

✓ The hub's private key has no password, so installing the hub's certificate works only from the hub's own computer - which this is.

Drop the response zip here, or [choose it](#)

Chipkin BACnet SC Hub 1.1.1 Chipkin Automation Systems Support: support@chipkin.com

Figure 6-3. Set-up guide step 3, Install the CA's response: the upload of the CARL response zip.

Step 4: Connect devices. Figure 6-4 lists the URI(s) devices connect to, the issuer certificate to download (`iss-1.pem`, `iss-1.cer` for Windows tools, `iss-2.pem`), and which file goes into which Network Port property of the device:

- Operational_Certificate_File: the device's `opr-<name>.pem` from the CA;
- the private key the device made with its CSR (it stays on the device);
- Issuer_Certificate_Files: `iss-1.pem` (in both slots if there are two);
- the primary hub URI.


BACnet SC Hub
Status
Set-up
About

CERTIFICATES

Chipkin BACnet SC Hub set-up guide

Chipkin BACnet SC Hub · v1.1.1 · device 389022

1 This hub's certificate
2 Get it signed
3 Install the CA's response
4 **Connect devices**
File inspector
Why did a connection fail?
Support report

Connect devices

Each device gets its own certificate from the **same CA** as the hub: send the CA the device's CSR (or a CARL request), and install what comes back on the device. The hub trusts every certificate that CA signs, right away - nothing to add on the hub.

Devices dial

wss://192.168.1.10:4443/ [copy](#) (this computer on the LAN)

wss://127.0.0.1:4443/ [copy](#)

The issuer certificate

Devices that need it use this to trust the hub. Load it into the device's issuer certificate (both slots, if it has two). It is public.

[Download iss-1.pem \(PEM\)](#)
[Download iss-1.cer \(DER, Windows\)](#)

The device's BACnet/SC Network Port settings

Operational certificate Operational_Certificate_File	the device's opr-<name>.pem from the CA
Private key	the key the device made with its CSR (it stays on the device)
Issuer certificate(s) Issuer_Certificate_Files	iss-1.pem (in both slots if there are two)
Primary hub URI	wss://192.168.1.10:4443/

Then start the device's connection and watch it on [Why did a connection fail?](#)

Figure 6-4. Set-up guide step 4, Connect devices: the hub URI, the issuer certificate downloads, and which file goes into which Network Port property.

Three tools are always available.

File inspector. Drop in any certificate, CSR, key, `.pfx` (empty password), or CARL zip. The file inspector, in Figure 6-5, shows every field and checks it against this hub; nothing is stored. The same inspector runs on the command line as `Chipkin-BACnet-SC-Hub --inspect <file>`, which exits with 1 if a check fails.



CAUTION

Don't upload a production private key to the file inspector or to any other tool.

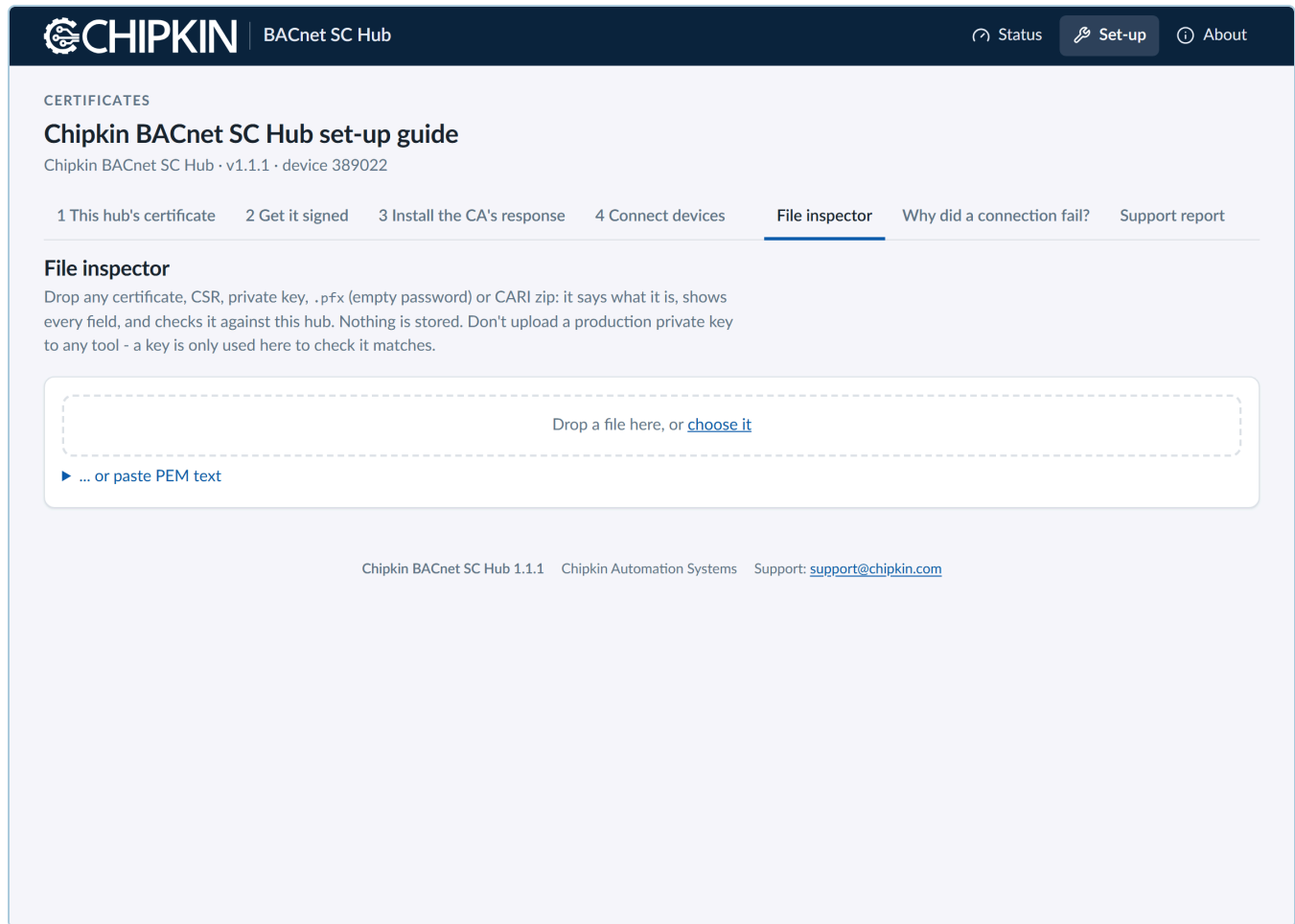


Figure 6-5. The set-up guide's File inspector tab.

Why did a connection fail? Start the device's BACnet/SC connection. The tab in Figure 6-6 shows, live, what the hub sees, and turns each failure into a sentence and a fix:

- a TCP connection with nothing after it: the device sent no certificate, only speaks TLS 1.2, or doesn't trust the hub's certificate;
- a refused certificate: OpenSSL's reason, such as signed by another CA, expired, not yet valid, no clientAuth, or revoked;
- the wrong WebSocket subprotocol;
- a refused Connect-Request (duplicate virtual MAC address (VMAC), missing Hello, the connection limit);
- and finally, "Connected!" with the device's VMAC and universally unique identifier (UUID).

CHIPKIN | BACnet SC Hub

STATUS Set-up ABOUT

CERTIFICATES

Chipkin BACnet SC Hub set-up guide

Chipkin BACnet SC Hub · v1.1.1 · device 389022

1 This hub's certificate 2 Get it signed 3 Install the CA's response 4 Connect devices File inspector Why did a connection fail? Support report

Why did a connection fail?

Start the device's BACnet/SC connection now. This page shows what the hub sees, live, and why an attempt failed.

Connected right now

- connected ○ = Example Building Controls, CN = AHU-1 controller 192.168.1.26:5646 VMAC 02:13:a8:27:eb:65 UUID a3f1b599-83f1-96d3-3ce3-0bac2ceaa281
- connected ○ = Example Building Controls, CN = Chiller plant controller 192.168.1.27:5646 VMAC 02:ab:20:ce:a2:81 UUID 1d49bf6a-8c64-e080-14f2-3d714b4b7c3b
- connected ○ = Example Building Controls, CN = Operator workstation 192.168.1.28:5646 VMAC 02:b7:fd:52:f6:3b UUID b1147923-5cf0-1ad4-5739-8a8b42496f7b

What the hub saw (newest first) live Only new

- 2026-10-13T03:35:23Z disconnected
"O = Example Building Controls, CN = AHU-1 controller" disconnected (192.168.1.20:5546).
close code 0 (no close frame: the connection dropped)
- 2026-10-13T03:35:22Z disconnected
"O = Example Building Controls, CN = Chiller plant controller" disconnected (192.168.1.24:5547).
close code 0 (no close frame: the connection dropped)
- 2026-10-13T03:35:22Z refused
The hub refused the device's BACnet/SC Connect-Request.
Connect-Request refused: "Hub function peer table full", error class 3 code 19 (VMAC 02:14:4c:1c:e8:ce)
- 2026-10-13T03:35:22Z connected
TLS worked: the hub accepted the certificate of "O = Example Building Controls, CN = Chiller plant controller" (192.168.1.24:5547).

Figure 6-6. The set-up guide's Why did a connection fail? tab.

Support report. A text summary to send to support: versions, the hub URI, the certificates with their checks, connected devices, and recent connection attempts. It contains no keys or passwords.

6.2 Who may install a certificate

Installing the CA's response (set-up guide step 3) needs the **password of the hub's private key**: the configuration file's `sc-key-password`, or the password typed when the hub started (see section 4, "Password-protected private keys"). With a key that has no password (what `--generate-csr` makes), it works only on the hub's own computer (from 127.0.0.1). Everything else in the guide is open, like the status page. See section 9, "Web pages".

6.3 Setting up a device

Most BACnet/SC devices and tools, such as the [Chipkin BACnet Explorer](#), take the files directly:

- **A tool that imports CARI files** takes the CA's response for the device as it is.
- **Any other tool or device:** unzip the response and install `opr-<name>.pem` (in `cert1/device-<instance>/port-<id>/`) and the device's own private key as its operational certificate and key, and `cert1/issuer/iss-1.pem` as its issuer certificate, in both slots if it has two. Set its primary hub URI to `wss://<hub address>:4443/`.

The **Chipkin BACnet Explorer** imports the device's CARL zip as it is (Connect, BACnet/SC, **Import...**). It connects through the network adapter you select, so give it the hub's **LAN address** (`wss://192.168.1.20:4443/`), not `wss://127.0.0.1:4443/`, even when it runs on the hub's own computer. Its "Test connection" checks each step and says which one failed.

YABE reads its certificate and key from a `.pfx` file. Make one from the device's files with an empty password, for example `openssl pkcs12 -export -in opr-yabe.pem -inkey key-yabe.pem -certfile iss-1.pem -passout pass: -out yabe.pfx`, keep it private, and give YABE `iss-1.cer` (from the set-up guide) as its trusted issuer. YABE leaves out the Hello option, so start the hub with `--sc-accept-device-without-hello on`. YABE uses Windows' own TLS, so the Windows version of the computer it runs on matters: see the TLS 1.3 item in section 10, "Devices that can't connect".

6.4 The connection audit trail

The hub logs each refused TLS handshake with the reason. It also keeps an audit trail of every connection, in lines that start with `SC audit:`. Each entry gives the device's address and certificate subject, then its VMAC and UUID once accepted.

```
2026-10-07 18:10:21 [INFO] SC audit: peer "wss://0.0.0.0:4443/|client=3" connected from 10.0.0.31:64150,
certificate "O = Example Site, CN = AHU-3 controller"
2026-10-07 18:10:21 [INFO] SC audit: peer "wss://0.0.0.0:4443/|client=3" (10.0.0.31:64150) is BACnet/SC
device VMAC 0a:0b:0c:0d:0e:0f, UUID 50515253-5455-5657-5859-5a5b5c5d5e5f, certificate "O = Example Site,
CN = AHU-3 controller" - connected
```

The same identity is logged again when the device disconnects. The status page lists the devices connected right now the same way.

6.5 Check that it works

- The devices appear under **Connected devices** on the status page.
- Every item in the status page's health checklist passes.
- From a BACnet/SC tool, a Who-Is finds the hub's device (instance 389022 by default).

7 Managing certificates over BACnet

A BACnet client can change the hub's certificates with the BACnet/SC certificate procedures (Standard 135-2024, clause 19.8.3), for example from the [Chipkin BACnet Explorer's](#) certificate page:

- 1. Write.** Send WriteProperty `File_Size = 0`, then AtomicWriteFile the new PEM certificate: File 1 for the hub's own certificate, File 3 or 4 for an issuer.
- 2. Check that it is staged.** Reading the File back shows the new certificate and Network Port 2's `Changes_Pending` is TRUE, but nothing is saved yet and the hub keeps using its current certificates.
- 3. Activate.** Send ReinitializeDevice `ACTIVATE_CHANGES` (or `WARMSTART`).

On activation the hub checks the staged certificates first: each must be a PEM certificate, at least one issuer must remain, and the hub's certificate must match its private key and chain to an issuer. If they pass, it saves them and restarts BACnet/SC; devices reconnect under the new certificates. If not, it responds with `INVALID_CONFIGURATION_DATA` and nothing changes, so a mistake can't lock the hub out. The staged writes stay, so the client can correct them and activate again.

If `dcc-password` is set, ReinitializeDevice requires it.

NOTE

Right after the hub starts, Network Port 2 reads `Changes_Pending` = TRUE (and `Current_Health` may report `invalid-configuration-data`) although nothing has been written. The CAS BACnet Stack leaves the hub's own start-up certificate settings pending. The certificates in use are correct. The first ReinitializeDevice `ACTIVATE_CHANGES` clears it.

7.1 Supported procedures

- **Add an issuer.** Write the new CA into slot 2 (File 4). Devices signed by either CA are then accepted, which lets a site move to a new CA gradually.
- **Replace the hub's certificate.** Read the CSR (File 2), have your CA sign it, write the result to File 1, and activate.
- **Replace the hub's certificate with a new key pair.** See the procedure in this section.
- **Discard staged writes.** Write `DISCARD_CHANGES` (1) to Network Port 2's `Command`. The staged certificate writes are dropped, the File objects read the files on disk again, and `Changes_Pending` goes back to FALSE. A restart also discards staged writes.

CAUTION

Don't send `DISCARD_CHANGES` to a freshly started hub. It also reverts the stack's start-up state, after which `Changes_Pending` no longer follows certificate writes (restart the hub to recover).

7.2 Replacing the hub's certificate with a new key pair

NOTE

A CSR made with `GENERATE_CSR_FILE` asks for no subjectAltName. If your devices check the hub's host name, renew with `--generate-csr` instead (see section 4, "Renewing the hub's certificate").

1. Write `GENERATE_CSR_FILE` (9) to Network Port 2's `Command`.
2. Read the new CSR from File 2.
3. Have your CA sign it.
4. Write the certificate to File 1.
5. Send `ReinitializeDevice` `ACTIVATE_CHANGES`.

In step 1 the hub makes a new key pair and a new CSR with the same subject, and saves the new key as `key-hub-pending.pem`. It keeps using its current key and certificate until step 5, when the new key replaces `key-hub.pem`. If the hub stops between saving the new certificate and swapping the key, it finishes the swap at the next start.

The set-up guide's `hub-cari-request.zip` also carries the new CSR, and installing the CA's response there or with `--import-cari` finishes the swap the same way as steps 4 and 5.

`GENERATE_CSR_FILE` is refused with `INVALID_VALUE_IN_THIS_STATE` while certificate writes are staged. It is also refused on a freshly started hub, which reads `Changes_Pending` `TRUE` with nothing staged: send `ReinitializeDevice` `ACTIVATE_CHANGES` once first.

8 Status page

The hub serves its web pages over HTTP on `--http-port` (default 8080), on `127.0.0.1` only by default, so the status page is at <http://127.0.0.1:8080/>. With `--http-tls` it serves HTTPS instead (TLS 1.2 or 1.3), using the hub's own certificate and key (`opr-hub.pem`, `key-hub.pem`) unless `--http-tls-cert` and `--http-tls-key` name another certificate, for example one your browsers already trust. The HTTPS certificate is loaded at start-up.

The status page shows the **connected devices** first, one row each: status, address, VMAC, device UUID, how long it has been connected (and since when, UTC), and its certificate's subject. Below them are the health checklist (hidden until you click its heading; its summary of passed, warnings, and failed always shows), the system health figures (BACnet/SC state, connections, certificates), and the traffic counters. A banner says when a newer version is available. Figure 8-1 shows the status page with three devices connected.

The screenshot shows the Chipkin BACnet SC Hub status page. The header includes the Chipkin logo, the text 'BACnet SC Hub', and navigation links for Status, Set-up, and About. The main content area is titled 'BACNET/SC HUB STATUS' and 'Chipkin BACnet SC Hub', with a version and update frequency note: 'Version 1.1.1 · device instance 389022 · updates every 5 s'.

Connected devices

Status	Address	VMAC	Device UUID	Connected	Certificate subject
connected	192.168.1.26:5646	02:13:a8:27:eb:65	a3f1b599-83f1-96d3-3ce3-0bac2ceaa281	0m 51s since 2026-10-13 03:34:33 UTC	O = Example Building Controls, CN = AHU-1 controller
connected	192.168.1.27:5646	02:ab:20:ce:a2:81	1d49bf6a-8c64-e080-14f2-3d714b4b7c3b	0m 48s since 2026-10-13 03:34:36 UTC	O = Example Building Controls, CN = Chiller plant controller
connected	192.168.1.28:5646	02:b7:fd:52:f6:3b	b1147923-5cf0-1ad4-5739-8a8b42496f7b	0m 48s since 2026-10-13 03:34:36 UTC	O = Example Building Controls, CN = Operator workstation

> Health checklist 6 passed

System health

Health	ok
Hub function (listener)	listening on wss://0.0.0.0:4443/
Hub connections (current / max)	3
Hub connector	off
Data link	BACnet/SC only (Network Port 2)

Traffic

Connects (total)	33
Disconnects (total)	30
Rate-limit rejections	0
RX	33 messages, 1122 bytes
TX	33 messages, 1247 bytes
Closed with a full transmit	0

Figure 8-1. The status page with three devices connected.

The page's figures update in place every 5 seconds without reloading the page, so it doesn't flicker or lose your place. Without JavaScript it reloads every 5 seconds instead.

If the hub stops responding, the page says **The hub is not responding** until it responds again.

Monitoring tools can read two addresses on the same port. `/health` reports *is the hub working?*: HTTP 200 with `status` `ok` when it is, and HTTP 503 with `status` `degraded` when the BACnet/SC hub isn't listening, usually because the hub has no certificate yet or a certificate is bad. `/metrics` reports *how much is it doing?* with counters such as uptime, connected devices, and messages and bytes in and out.

8.1 The health checklist

Beside the system health figures, the status page lists the checks the hub makes, each with an icon and the word **Pass**, **Warning**, or **Fail**, and for anything but a pass, what to do (with a link to the set-up guide where that helps). Figure 8-2 shows the checklist expanded.

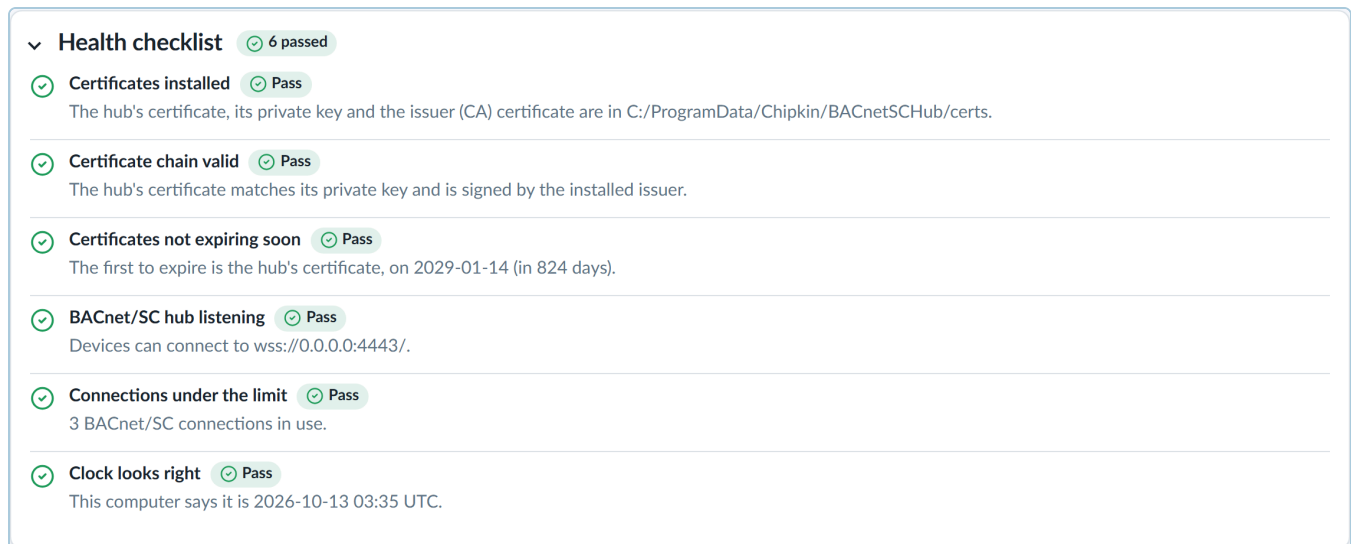


Figure 8-2. The status page's health checklist, expanded.

CHECK	PASSES WHEN
Certificates installed	The hub's certificate, its private key, and an issuer certificate are in the certificate folder.
Certificate chain valid	The hub's certificate matches its private key, isn't a CA certificate, and is signed by an installed issuer for both TLS roles (the same check as an activation over BACnet). A staged change made over BACnet is a warning until it is activated.
Certificates not expiring soon	The hub's and the issuers' certificates are valid now and for more than 30 days (a warning within 30 days; a failure once expired, or not yet valid).
BACnet/SC hub listening	The hub function is listening for devices.
Connections under the limit	Fewer devices are connected than the limit (a warning when full: the next device is refused).
Clock looks right	The computer's clock is not earlier than the day this version was built (certificates are checked against it).
TLS key log off	Shown only while <code>--sc-keylog-file</code> is on, as a warning.
Software up to date	Shown once the update check has a result: a pass when this is the latest version, a warning with the download link when a newer one is available. Not shown while the check is off or couldn't reach the update service.

8.2 Update check

At start-up and every 24 hours the hub asks Chipkin's update service whether a newer version exists. It runs in the background, so the hub never waits for it, and gives up after about 10 seconds.

- **A newer version:** a yellow `UPDATE AVAILABLE` line on the console (and in the log file) with the download link, a banner on the status page, and a **Software up to date** warning in the health checklist. The hub **never downloads or installs anything**. Install the new version when it suits you.
- **Up to date:** one log line, and the checklist item passes.
- **No reply** (an offline or air-gapped site, a firewall, a proxy): one quiet `INFO` line saying it couldn't check. Nothing else changes.

What is sent: one HTTPS request. Nothing about your site, devices, certificates, or settings is sent, and no cookie. Like any web request, it shows Chipkin's service the public IP address of your site.

Checking now: the About page's **Check for updates** button asks the update service at once instead of waiting for the next check, and shows the result: up to date, a newer version with its download link, couldn't reach the service, or turned off. It works from the hub's own computer only (a page opened as `http://127.0.0.1` or `localhost`), at most once every 30 seconds.

Turning it off: `--no-update-check` on the command line, or `check-for-updates = false` in the configuration file. The About page then says the check is off.

8.3 About and feedback

The **About** page (in the app bar of every page) shows the hub's version, the CAS BACnet Stack version, the build (the commit it was built from and the platform), the operating system, and the device, and has the **Check for updates** button. Its figures stay current like the status page's. Figure 8-3 shows the About page.

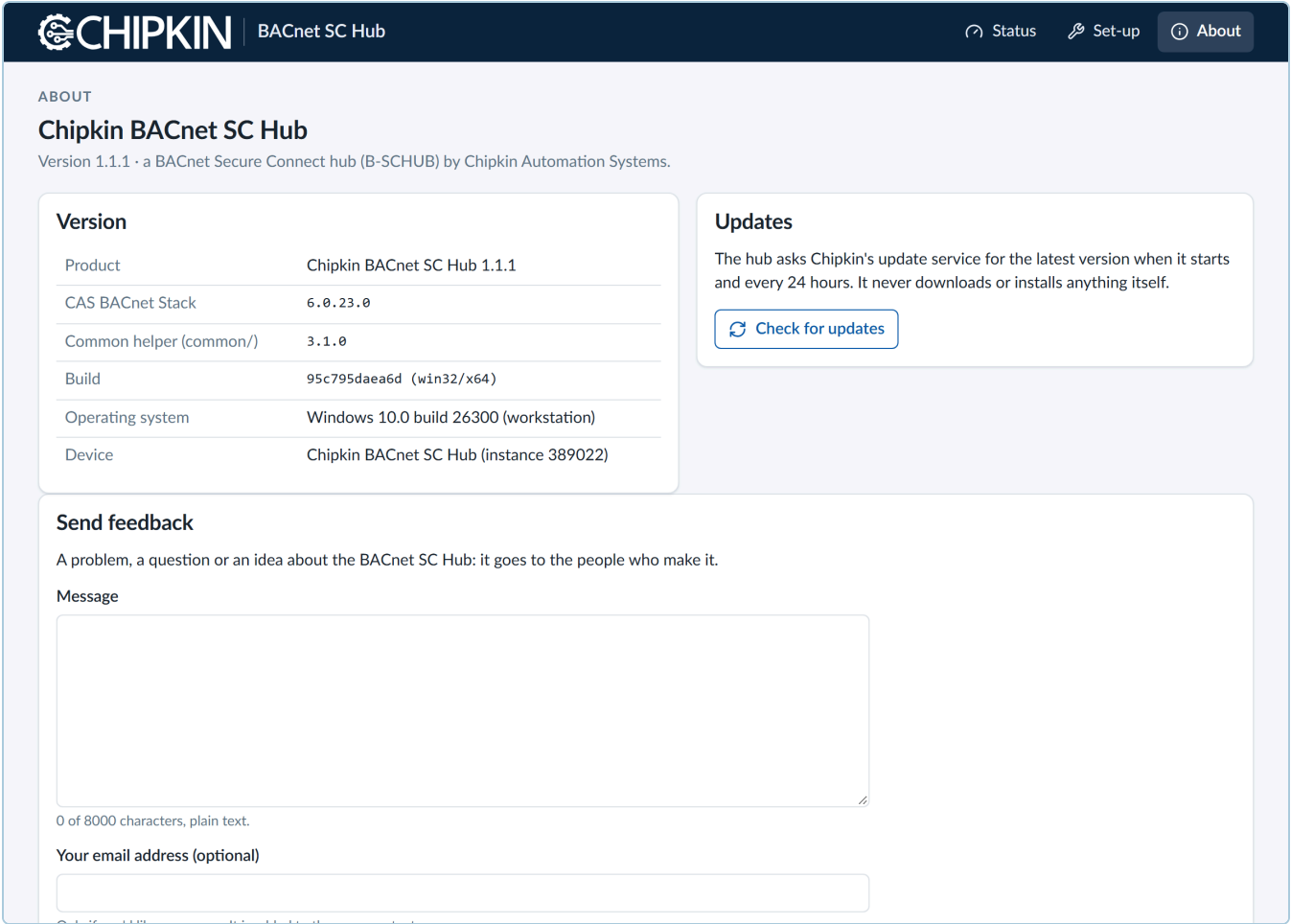


Figure 8-3. The About page.

Its **Send feedback** form sends a message to the people who make the hub: a problem, a question, or an idea. A message is plain text up to 8000 characters. Your email address is optional; give it if you would like a reply. Tick **Include diagnostics** to add the hub's version, operating system, CAS BACnet Stack version, uptime, connection count, and the health checklist's results; **Show exactly what is added** shows the text before you send it. No certificates, keys, passwords, addresses, hub URLs, folder names, or device names are ever included.

Nothing is sent until you press **Send feedback**. The message goes to Chipkin over HTTPS, with the hub's version. If the hub can't reach Chipkin's service, the page says so; email support@chipkin.com instead.

9 Security

9.1 BACnet/SC connections

BACnet/SC uses TLS 1.3 only, with mutual authentication, and every BACnet message to or through the hub is BACnet/SC. A device is accepted if its certificate chains to an issuer in File 3 or 4 and isn't revoked.

Host names aren't checked when the hub connects out to another hub. BACnet/SC certificates identify devices, not Domain Name System (DNS) names; the certificate chain is still verified.

New BACnet/SC connection attempts are limited per source address (`--sc-rate-limit`, default 10 a second) and for the whole listener (`--sc-rate-limit-total`, default 50 a second), before the TLS handshake. One flooding host uses up only its own allowance, so devices on other addresses can still reconnect. The hub logs each refusal with the address. A device that stops reading is disconnected after 64 messages are waiting for it.

`--sc-keylog-file` writes the BACnet/SC session secrets to a file, so that anyone with the file and a capture can read the traffic. It's for debugging only, and is off unless given on the command line (see section 10, "Decoding BACnet/SC traffic in Wireshark").

9.2 Certificate revocation

Put your CA's CRLs in `issuer-crl.pem` in `--sc-cert-dir` (PEM; one CRL per issuer, concatenated). The hub then refuses any device whose certificate is on a list. It checks the file every 5 seconds: a new or changed file restarts BACnet/SC, every device reconnects, and revoked ones are refused, so no restart is needed. Without the file, revocation isn't checked. The start-up log lists each CRL, how many certificates it revokes, and when it expires.

9.3 Replacing a device and revoking its certificate



CAUTION

Once you use `issuer-crl.pem`, keep its CRLs current. Revocation checking fails closed: a device whose issuer's CRL is missing or past its next-update date is refused, and a file with no readable CRL stops BACnet/SC.

1. Get the replacement device its own certificate from your CA and connect it (see section 6, "Connecting devices").
2. In your CA, revoke the old device's certificate.
3. In your CA, make a new revocation list. The Chipkin BACnet SC Certificate Authority saves it as `issuer-crl.pem`.
4. Put `issuer-crl.pem` in the certificate folder of every hub, the primary and the failover.
5. Wait a few seconds for the hub to restart BACnet/SC.
6. Confirm on the status page that the other devices appear under **Connected devices** again.

9.4 Web pages

The web pages are served on 127.0.0.1 by default. If you need them from another computer, turn on `--http-tls` before `--http-bind`, so the status data and anything you send to the hub are encrypted (or use an SSH tunnel or a TLS reverse proxy). The hub logs a warning on every start bound off loopback over plain HTTP. The status page and

the monitoring addresses have no authentication, even over HTTPS.

Installing a certificate from the set-up guide needs the hub key's password. Without a password on the key, it works only on the hub's own computer. From another computer it also needs `--http-tls`. Every installation and refusal is logged.

The status page, the set-up guide, and the About page load nothing from elsewhere. They carry their styles, logo, and icons inside the page and are served with a strict Content-Security-Policy, so they work offline and can't pull in outside content.

9.5 Private keys

Private keys (`key-hub.pem`, `key-hub-pending.pem`, and any device key or `.pfx`) must stay private. The hub's request zip never contains its key, and the private key is never served by any BACnet object.

The hub never signs certificates. It holds no CA key, so taking over the hub doesn't let anyone issue certificates for your BACnet/SC network. Your CA does that.

9.6 Connections to Chipkin

The hub connects out to Chipkin's update service (`https://app-updates.chipkin.com`) for two things only: the update check (start-up and every 24 hours; turn it off with `--no-update-check` or `check-for-updates = false`) and a message someone sends from the About page. Both use HTTPS with the service's certificate checked; neither sends certificates, keys, or settings, and the hub never downloads or runs anything from the service. See section 8, "Update check" and "About and feedback".

To report a security problem, see section 13, "Contact us for support".

10 Troubleshooting

10.1 Start-up and certificates

SYMPTOM	CAUSE	WHAT TO DO
<code>cannot start listening ... certificate file(s) missing/unreadable</code>	The hub has no certificate yet, so it is unreachable over BACnet.	Run <code>--generate-csr</code> , have your CA sign <code>hub-cari-request.zip</code> , and install the CA's response with <code>--import-cari</code> or in the set-up guide. The status page shows what to fix.
<code>--import-cari</code> or the set-up guide says the response is "for another request"	The CA signed a different CSR: an earlier request, or another hub's.	Send the CA this hub's current <code>hub-cari-request.zip</code> .
<code>--import-cari</code> or the set-up guide says "no issuer certificate in the zip signed the hub's certificate"	The response is missing the CA that signed it (or has only a root, and an intermediate signed the certificate).	Ask the CA for a response with the issuing CA in <code>cert1/issuer/</code> .
<code>lws_create_context failed ... retrying every 5 s</code>	The BACnet/SC port is in use, or a certificate file doesn't load.	Fix the cause. The hub keeps retrying and starts listening without a restart.
<code>private key ... DOES NOT MATCH</code> at start-up	The hub's key and certificate (<code>key-hub.pem</code> and <code>opr-hub.pem</code>) don't belong together.	Install the CA's response for this hub's current request.
The status page says the BACnet/SC hub isn't listening	BACnet/SC isn't listening, so the device is unreachable over BACnet.	Open the health checklist: it says which check fails and what to do.
BACnet/SC doesn't start with certificates from the public B-SCHUB example's CA	That CA's private key is public, so the hub never trusts it and refuses to import its certificates.	Get the hub's and the devices' certificates from your own CA.

10.2 Devices that can't connect

SYMPTOM	CAUSE	WHAT TO DO
A device won't connect, and you don't know why	The hub records what it saw of each attempt.	Open the set-up guide, <i>Why did a connection fail?</i> , and start the device. The file inspector (or <code>--inspect <file></code>) checks the device's certificate against the hub.
A device can't reach the hub	The hub URI or the firewall is wrong, or the hub already has its maximum number of devices connected (<code>--sc-max-hub-connections</code>).	Check the hub URI the device uses and TCP 4443 in the firewall. The <code>SC audit:</code> lines show what the hub saw.
<code>SC TLS handshake REJECTED - client certificate failed verification</code>	The device's certificate isn't signed by an issuer the hub trusts, has expired, or is revoked (certificate revoked, unable to get certificate CRL when <code>issuer-crl.pem</code> has no CRL for its issuer, CRL has expired).	Check it with <code>openssl verify -CAfile cert1/issuer/iss-1.pem <the device's opr-*.pem></code> (add <code>-crl_check -CRLfile issuer-crl.pem</code> with a CRL).
<code>refusing a WebSocket upgrade ... (HTTP 400)</code>	The device asked for a WebSocket subprotocol other than <code>hub.bsc.bacnet.org</code> .	Check the device's BACnet/SC settings.

SYMPTOM	CAUSE	WHAT TO DO
Connect-Request refused by the hub	Another connected device has the same VMAC, the device left out the Hello option, or the hub is full.	Give the device its own VMAC, see section 5, "BACnet/SC compatibility", or disconnect another device.
SC rate limit: refusing a new connection	A device (or something else) is reconnecting faster than <code>--sc-rate-limit</code> allows.	Find the device at the logged address and check why it keeps reconnecting.
The Chipkin BACnet Explorer's test stops at "Reach the hub's port": "The local address could not be used"	It binds to the network adapter selected on its Connect page and can't reach <code>127.0.0.1</code> from there.	Use the hub's LAN address in the hub URI.
A BACnet/SC tool on a Windows 10 computer (for example YABE) fails with <code>A call to SSPI failed</code> or a TLS handshake error, and the hub logs nothing	The tool uses Windows' own TLS, which on Windows 10 can't make the TLS 1.3 connections BACnet/SC requires, to this hub or to any other BACnet/SC hub.	Run the tool on Windows 11, Windows Server 2022, or later, or use a client with its own TLS 1.3 (Chipkin BACnet Explorer).

10.3 Web pages and console

SYMPTOM	CAUSE	WHAT TO DO
The set-up guide won't install the response from another computer	The hub key has no password, or the pages are served over plain HTTP.	Open the guide on the hub's own computer (127.0.0.1), or give the key a password, set <code>sc-key-password</code> , and start the hub with <code>--http-tls</code> .
The status page doesn't open in a browser	It opens only when the hub is started from a console; on Linux only in a desktop session.	Open the address the hub prints.
Escape codes like <code>[33m</code> in a console	Your console doesn't show colour.	Set <code>NO_COLOR=1</code> .

10.4 Decoding BACnet/SC traffic in Wireshark

BACnet/SC is TLS 1.3, so a capture shows only encrypted records. To see the BACnet/SC messages inside, have the hub write each connection's TLS session secrets to a key log file, and give that file to Wireshark:

```
Chipkin-BACnet-SC-Hub --sc-keylog-file sc-keys.log
```



WARNING

Anyone with this file and a capture can read that traffic. Use the option for debugging only, then delete the file.

1. Start the capture (for example `tcp port 4443`; on Windows, capture traffic to the same computer with Npcap's loopback adapter).
2. Start the hub with `--sc-keylog-file`. Only connections made while the option is on can be decoded, because Wireshark needs each TLS handshake.
3. In Wireshark, open **Edit > Preferences > Protocols > TLS > (Pre)-Master-Secret log filename**, and select `sc-keys.log`. With tshark: `tshark -r capture.pcapng -o tls.keylog_file:sc-keys.log`.

4. Read the decoded records. They decode as WebSocket, and the BACnet/SC frames as **BSCVLC** (Connect-Request, Connect-Accept, Encapsulated-NPDU, ...). Filter with `bscv1c`. If you run BACnet/SC on a port Wireshark doesn't recognize, use **Decode As > TCP port > TLS**.

The file covers both BACnet/SC roles: devices connecting to the hub, and the hub's own connection to another hub (`--sc-hub-uri`). It doesn't cover the HTTPS status page. The hub appends to the file and flushes every line straight away, so you can decode a capture while the hub is still running.

The file never holds a private key. The option can be set on the command line only (there is no configuration key), so a service can't be left logging by a forgotten line in its configuration file. While it is on, the hub logs a warning at start-up and whenever BACnet/SC starts listening, and the status page shows a banner.

11 The BACnet device

Device 389022	"Chipkin BACnet SC Hub"	Vendor 389 (Chipkin Automation Systems)
└─ Analog Input 1	"Bronze"	sample value, degrees C
└─ Network Port 2	"BACnet SC"	BACnet/SC hub function (and optional connector)
└─ File 1	"Operational Certificate"	the hub's certificate (writable)
└─ File 2	"Certificate Signing Request"	CSR for the hub's key (read-only)
└─ File 3	"Issuer Certificate Slot 1"	trusted CA certificate (writable)
└─ File 4	"Issuer Certificate Slot 2"	second trusted CA certificate (writable)

Every object has a **Description** saying what it is for.

The device instance defaults to 389022 (`--deviceID`), and the Device's name to "Chipkin BACnet SC Hub" (`--device-name`). Both must be unique on the BACnet internetwork. Analog Input 1 is a sample value: the up and down arrow keys change it while the hub runs.

Network Port 2 (BACnet/SC) is the device's only Network Port (see section 5, "BACnet/SC only"). It reports `Network_Number` 0 with `Network_Number_Quality` **unknown** unless `--sc-network-number` sets it; this device isn't a router. The PICS lists every property.

12 Version and licensing

- **Version.** `Chipkin-BACnet-SC-Hub --version` prints the version you are running, with the CAS BACnet Stack version it is built on. What a version number promises, and how long a version is supported, is in the Support policy: each MAJOR version gets security fixes for 5 years from its first release.
- **Licensing.** The Chipkin BACnet SC Hub is proprietary software of Chipkin Automation Systems Inc. (see `LICENSE`). Licensing is arranged with Chipkin: use <https://store.chipkin.com/contact-us> or email sales@chipkin.com. The hub is built on the **CAS BACnet Stack**, a separate Chipkin product, and on OpenSSL (Apache-2.0), libwebsockets (MIT), libuv (MIT), zlib (Zlib), and, on Windows, pthreads4w (Apache-2.0). Its web pages use Lucide icons (ISC). See the Third-party notices. Each release's `sbom-*.cdx.json` lists the exact versions.
- **Related documents.** The fact sheet, the PICS, the Production certificates guide, and the Support policy.

13 Contact us for support

Chipkin provides technical support for the Chipkin BACnet SC Hub.

- **Email:** support@chipkin.com
- **Phone:** 1-866-383-1657 (toll free)
- **Web:** <https://store.chipkin.com/contact-us>

When you report a problem, include:

- the output of `Chipkin-BACnet-SC-Hub --version`;
- the operating system;
- the relevant log lines;
- for connection problems, the `SC audit:` and `SC TLS` lines.

The set-up guide's Support report has all of this and no keys or passwords, so you can send it instead.

For sales and licensing, email sales@chipkin.com.

To report a security problem, email support@chipkin.com privately with "Security" in the subject.

A Document history

This table lists the released versions of this manual.

DOCUMENT REVISION	DATE	PRODUCT VERSION	CHANGES
1.0	2026-Oct-15	1.1.1	First release.

B Example configuration

This is a complete configuration file for an installed hub on Windows. Every key is optional: a key you leave out keeps its default. Relative paths are relative to the folder you start the hub from, which for an installed hub is the folder that holds the file.

```
# Device
device-id = 389022
device-name = Example Site BACnet SC Hub
ip-address = 192.168.1.20

# BACnet/SC
sc-port = 4443
sc-cert-dir = certs
sc-network-number = 100
sc-max-hub-connections = 4
sc-rate-limit = 10
sc-rate-limit-total = 50
sc-accept-hub-without-hello = false
sc-accept-device-without-hello = false

# Optional hub connector: also connect out to another BACnet/SC hub
# sc-hub-uri = wss://192.168.1.30:4443/
# sc-failover-uri = wss://192.168.1.31:4443/

# Passwords (configuration file only; restrict this file's permissions)
# dcc-password = <1 to 20 characters>
# sc-key-password = <the password of the hub's private key>

# Web pages
http-port = 8080
http-bind = 127.0.0.1
http-tls = false
# http-tls-cert = <file>
# http-tls-key = <file>

# Logging
log-file = logs\hub.log
log-max-size-mb = 10
log-max-files = 5

# Update check
check-for-updates = true
```

An installed hub reads `C:\ProgramData\Chipkin\BACnetSCHub\hub.conf` on Windows and `/etc/chipkin-bacnet-sc-hub/hub.conf` on Linux.

C Glossary of terms

TERM	MEANING
BACnet Secure Connect (BACnet/SC)	The BACnet data link that carries BACnet over WebSockets secured with TLS (Standard 135-2024, Annex AB).
BIBB	BACnet Interoperability Building Block: a named set of BACnet services that a device supports, such as NM-SCH-B.
CA (certificate authority)	The application or service that signs certificates; the hub and its devices trust the certificates their CA signed.
CARI	Certificate Authority Requirements Interchange: the folder and zip layout BACnet uses to take CSRs to a CA and bring certificates back (Standard 135-2024, Annex AA.2).
Certificate	A file, signed by a CA, that identifies a hub or a device to the other side of a TLS connection.
CRL (certificate revocation list)	A list from the CA of certificates that are no longer to be accepted.
CSR (certificate signing request)	A file that holds a public key and a name, which a CA signs to make a certificate.
Device instance	The number that identifies a BACnet device on the internetwork; the hub's is 389022 by default.
Failover hub	A second hub that devices connect to when the primary hub is down.
Hub	The BACnet/SC device that every other device connects to and that relays their traffic.
Hub connector	The part of a BACnet/SC device that connects out to a hub; this hub has an optional one, turned on with <code>--sc-hub-uri</code> .
Issuer certificate	The certificate of the CA that signed a device's or hub's certificate (<code>iss-1.pem</code> , <code>iss-2.pem</code>).
Network Port	The BACnet object that represents a data link of a device; the hub's only one is Network Port 2, "BACnet SC".
Operational certificate	The certificate a hub or device presents on its BACnet/SC connections (<code>opr-<name>.pem</code>).
PEM	Privacy-Enhanced Mail: the text file format the hub uses for certificates, CSRs, and keys.
PICS	Protocol Implementation Conformance Statement: the document that lists the objects, properties, and services a BACnet device supports.
Private key	The secret half of a key pair, which proves that a certificate belongs to its holder and must never leave it.
subjectAltName	The certificate field that lists the host names and addresses a certificate is valid for.
TLS	Transport Layer Security: the protocol that encrypts a connection and lets each side check the other's certificate.
UUID	Universally unique identifier: the fixed identifier a BACnet/SC device gives when it connects to a hub.
VMAC	Virtual MAC address: the address a device uses on a BACnet/SC network.
WebSocket	The connection protocol, started with an HTTP request, that BACnet/SC messages travel over.
<code>wss://</code>	The start of a WebSocket URI that uses TLS, as in the hub URI <code>wss://<hub address>:4443/</code> .