



Chipkin BACnet SC Certificate Authority CAS-1000-22

Chipkin - Enabling Integration

sales@chipkin.com

Tel: +1 866 383 1657

© 2026 CHIPKIN AUTOMATION SYSTEMS

Product version: 0.6.1

Document revision: 1.0

Issue date: 2026-Oct-15

Contents

1	What the Certificate Authority does	4
1.1	How it fits together	4
1.2	The steps, in order	5
1.3	Help in the app	6
2	Installation	7
2.1	Supported platforms	7
2.2	Installing	7
2.3	Starting the app and opening a CA file	7
2.4	Updates	8
3	Quick start	10
4	Working with an open CA	12
4.1	Saving and closing	12
4.2	Certificate information	13
4.3	The change log	13
5	Creating a site CA	14
5.1	Why the backup matters	16
5.2	A CA file with no password	16
6	Signing requests	17
6.1	The Chipkin BACnet SC Hub's request	17
6.2	Devices that make their own request	19
6.3	What the app checks	20
7	Making a device's key and certificate	22
7.1	Installing on the device	24
8	Revoking and renewing	26
8.1	Revoking a certificate and making the revocation list	26
8.2	Keeping the revocation list current	28
8.3	Renewing a certificate	29
8.4	Renewing a hub's certificate	29
9	The register	31
10	Files and passwords	32
10.1	Which files have a password	32
10.2	Every file and where it goes	32
10.3	The CA file	33
11	Checking any file	34

12	Moving to a new computer and backing up	36
12.1	Moving to a new computer	36
12.2	Backing up	36
12.3	If the CA file or its password is lost	36
12.4	Moving the CA to another tool	36
13	Privacy: what is sent and what is never sent	37
13.1	What is sent, and when	37
13.2	What is never sent anywhere	38
14	Routine upkeep	39
15	Troubleshooting	40
15.1	Installing and opening	40
15.2	Signing	40
15.3	Hubs and devices	40
16	Version and licensing	42
17	Contact us for support	43
A	Document history	44
B	Glossary of terms	45

1 What the Certificate Authority does

This manual is for people who set up and look after the certificates of a BACnet Secure Connect (BACnet/SC) site with the Chipkin BACnet SC Certificate Authority (part number CAS-1000-22). You don't need to know certificates already. Each term is explained the first time it matters.

The Chipkin BACnet SC Certificate Authority is a desktop application for Windows and Linux. It makes and signs the certificates a BACnet/SC network needs. Every device on a BACnet/SC network, and every hub the devices connect to, needs a certificate signed by the same certificate authority (CA). This manual calls the application "the app", and calls the signing identity you create with it for your site "the CA".

Everything happens on your computer. Your CA, keys, and files are never uploaded, and there is no account. The app asks Chipkin only whether there is a newer version (you can turn that off), and sends feedback only when you send it. See section 13, "Privacy: what is sent and what is never sent".

1.1 How it fits together

BACnet/SC (ANSI/ASHRAE Standard 135-2024) connects every device to a **hub** over Transport Layer Security (TLS) 1.3, and both sides prove who they are with a **certificate**. A hub accepts a device, and a device accepts the hub, only if the same **CA** signed both certificates. You make the CA once per site, then use it to sign the certificate of each hub and each device.

Figure 1-1 shows a BACnet/SC network on a corporate local area network (LAN). The Certificate Authority sits beside the network, not on it.

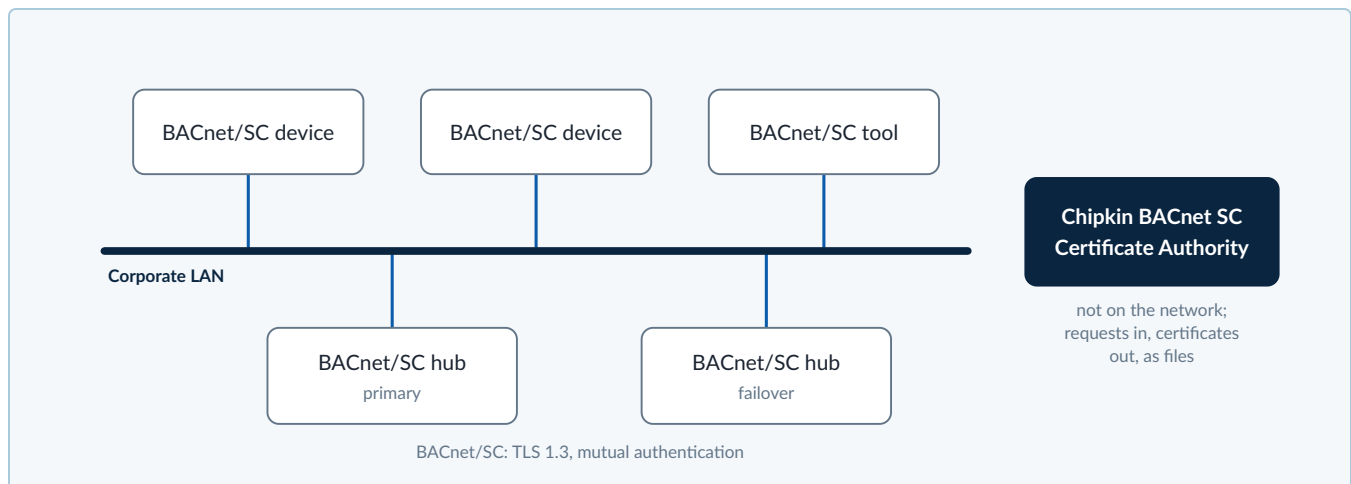


Figure 1-1. A BACnet/SC network on a corporate LAN, with the Certificate Authority not connected to the network.

The parts of the picture:

- **Private keys.** Each hub and device has its own private key. Normally it makes the key itself and the key never leaves it. It sends the CA a **request**, which is a certificate signing request (CSR). A request holds the public key and the name of the hub or device.
- **Certificates.** The CA checks the request and signs it. The result is the device's certificate (`opr-<name>.pem`). It goes back in a **response zip**, a zip file that holds the request, the certificate, and the CA's own certificate (`iss-1.pem`).

- **Trust.** Every hub and device loads its certificate, its key, and the CA's certificate. From then on they trust each other. Nothing has to be added on the hub for each new device.
- **Hubs.** Devices connect to the **primary hub**. A site can also have a **failover hub**, which devices connect to when the primary is down. Each hub needs its own certificate from the same CA.
- **No network needed.** The CA doesn't need the BACnet network, or any network. Requests come to it and certificates go back as files (on a USB stick, by email, or through a shared folder).

Requests and response zips use the Certificate Authority Requirements Interchange (CARI) format (Standard 135-2024, Annex AA.2). CARI is the folder layout BACnet uses to take requests to a CA and bring the certificates back. The app works with the Chipkin BACnet SC Hub (part number CAS-1000-21), the Chipkin BACnet Explorer, and any CARI-compatible tool.

Figure 1-2 shows the files that move between the hubs, the devices, and the app.

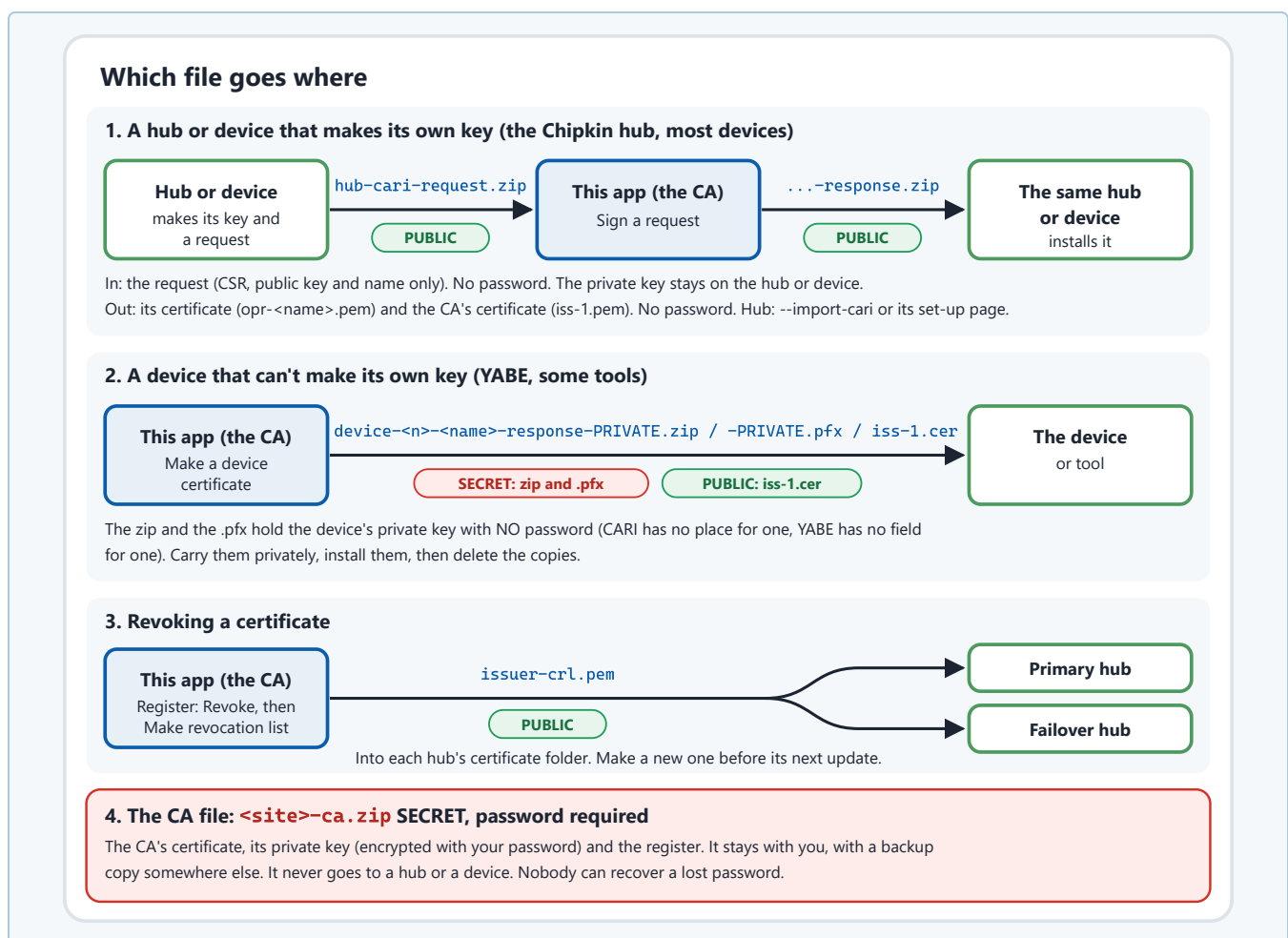


Figure 1-2. Which file goes where: requests come to the Certificate Authority, and responses and the revocation list go back to the hubs and devices.

1.2 The steps, in order

1. Create the CA and back it up.
2. Sign the primary hub's request.

3. Sign the failover hub's request, if the site has a failover hub.
4. Sign each device's request, or make the device's key and certificate in the app if the device can't make a request.
5. Save the CA file again.
6. When a device is lost or retired, revoke its certificate and give every hub the new revocation list.
7. Renew certificates before they expire.

Section 3, "Quick start", walks through steps 1 to 5.

1.3 Help in the app

The app has help on every screen:

- **Getting started** (at the top of the window) has the network and file pictures, the order to do things in, and the tables of files and passwords.
- **User guide** (beside it) opens the guide inside the app.
- Every screen has a **Help** panel. Click it to open it. It gives what the step does, why, which files go in and come out, and where each goes next.
- Each field has a description beside it.

To get around, click the Chipkin logo to go to the main page (CA home when a CA is open). The mouse's back button or Alt+Left returns to the previous screen. Some screens also have a **Back** button.

2 Installation

2.1 Supported platforms

- Windows 11 or later (64-bit)
- Linux (64-bit), as an AppImage or as a Debian or Ubuntu package

2.2 Installing

Download the installer for your computer from the release page Chipkin gave you.

COMPUTER	FILE
Windows 11 or later (64-bit)	Chipkin-BACnet-SC-Certificate-Authority-<version>-setup.exe
Linux (64-bit), any distribution	Chipkin-BACnet-SC-Certificate-Authority-<version>-x86_64.AppImage
Debian or Ubuntu	Chipkin-BACnet-SC-Certificate-Authority_<version>_amd64.deb

To install:

- **Windows:** Run the installer. Choose to install for yourself or for everyone, and choose the folder.
- **Linux, AppImage:** Make the file executable (`chmod +x` the file, or **Properties, Permissions**) and run it.
- **Debian or Ubuntu:** Run `sudo apt install ./Chipkin-BACnet-SC-Certificate-Authority_<version>_amd64.deb`.

i NOTE

Windows may show "Windows protected your PC" when you run the installer. Click **More info**, check that the file is the one from Chipkin, then click **Run anyway**.

i NOTE

On Ubuntu 24.04, if the AppImage doesn't start, install the `.deb` instead. Don't run the app with `--no-sandbox`.

2.3 Starting the app and opening a CA file

When the app starts, it shows the start screen (Figure 2-1). From there you create a new site CA, open a CA file, or inspect a file.

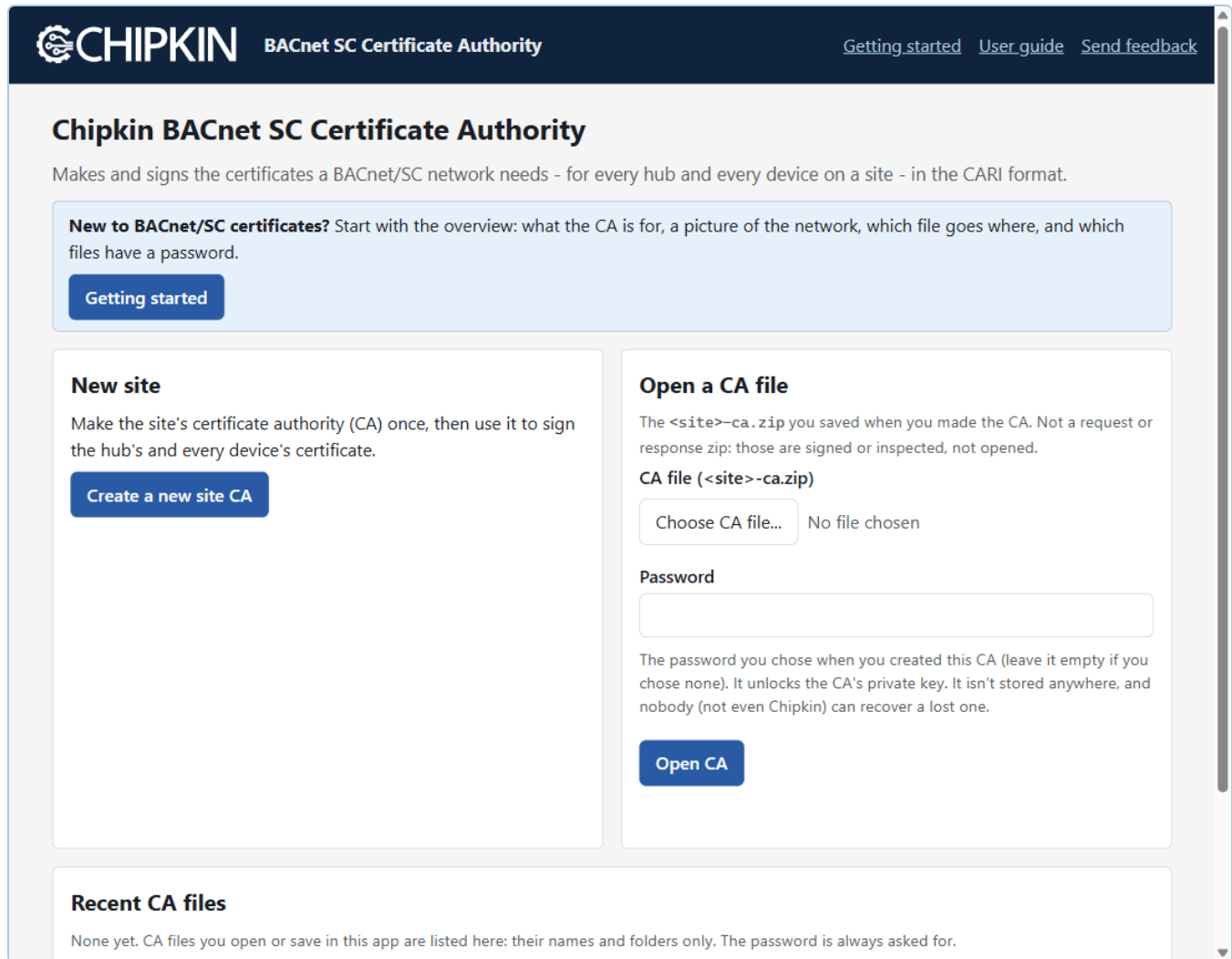


Figure 2-1. The start screen: Getting started, Create a new site CA, Open a CA file, and Recent CA files.

A CA file is a `.zip` file, so double-clicking it opens your zip tool, not the app. To open a CA file:

1. On the start screen, click **Choose CA file...** and choose the file, or click the file under **Recent CA files**.
2. Type the CA file's password in **Password**.
3. Click **Open CA**.

On Windows you can also right-click the CA file, choose **Open with**, and choose the app.

2.4 Updates

The Windows app and the Applmage check for a newer version when they start and every 24 hours. When there is one, a notice appears under the header (Figure 2-2) with two choices, **Download and install** and **Later**.

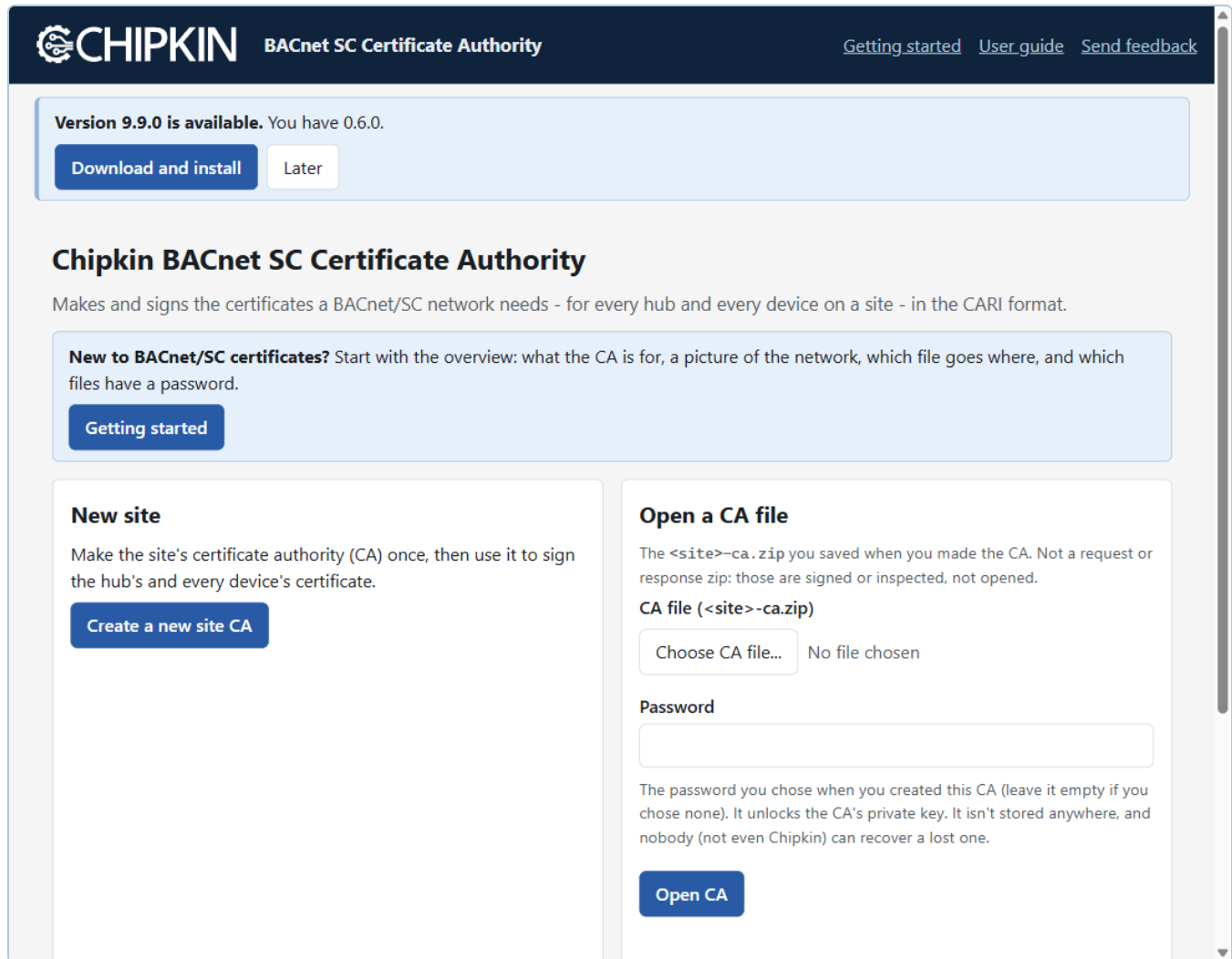


Figure 2-2. The update notice under the header, with Download and install and Later.

In the Windows app and the Applmage, **About and updates** (at the bottom of the window) has **Check for updates** and the **Check for updates automatically** box. Clear the box to turn the automatic check off.

Nothing is downloaded until you choose **Download and install**. The installer is checked against its published SHA-256 checksum (a Secure Hash Algorithm value that shows the file is unchanged) before it runs. The app then asks whether to install it now.

The `.deb` doesn't update itself. Install the new `.deb` over the old one. Your CA files and the list of recent CA files are kept. On a `.deb` install, **About and updates** shows only a message to install the new `.deb`.

On any platform you can update by installing the new version over the old one.

3 Quick start

These steps set up a new site so that its hub and devices trust each other. Each step points to the section with the detail. The **Getting started** page in the app (Figure 3-1) gives the same steps.

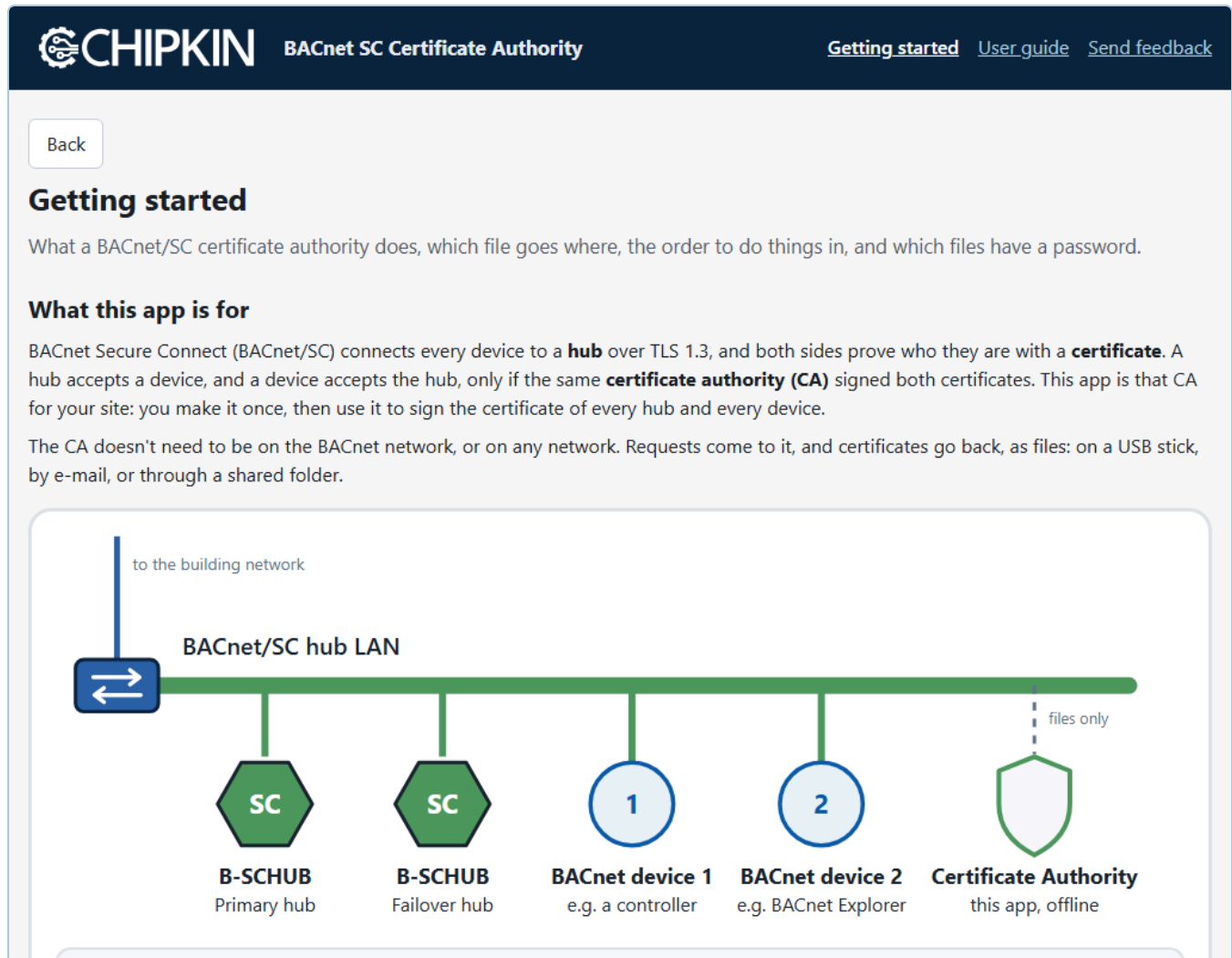


Figure 3-1. The Getting started page: what the app is for, with the network picture.

- 1. Create the CA.** On the start screen, click **Create a new site CA**, name the CA, choose a password for the CA file, and click **Create CA**. See section 5, "Creating a site CA".
- 2. Back it up.** Click **Save CA file**, then **Save a second copy** and save it somewhere else. Store the password in a password manager.
- 3. Sign the primary hub's request.** Copy `hub-car1-request.zip` from the hub's computer, click **Sign a request**, choose the file, and sign it. Copy the response zip back to the hub and install it. See section 6, "The Chipkin BACnet SC Hub's request".
- 4. Sign the failover hub's request** the same way, if the site has a failover hub. Each hub has its own key and its own certificate.

5. **Give each device its certificate.** If the device makes its own request, sign it (see section 6, "Devices that make their own request"). If it can't, click **Make a device certificate** (see section 7, "Making a device's key and certificate").
6. **Save the CA file again.** Click **Save** at the top of the window, or press Ctrl+S. Then copy the saved file over your backup.

Afterwards, revoke a certificate when a device is lost or retired, and renew certificates before they expire. See section 8, "Revoking and renewing", and section 14, "Routine upkeep".

4 Working with an open CA

Once a CA is open, the top of the window shows its name (click it for CA home), **Save** and **Close**, and the tabs **CA home**, **Sign a request**, **Make a device certificate**, **Inspect a file**, and **Issued certificates**. Figure 4-1 shows CA home.

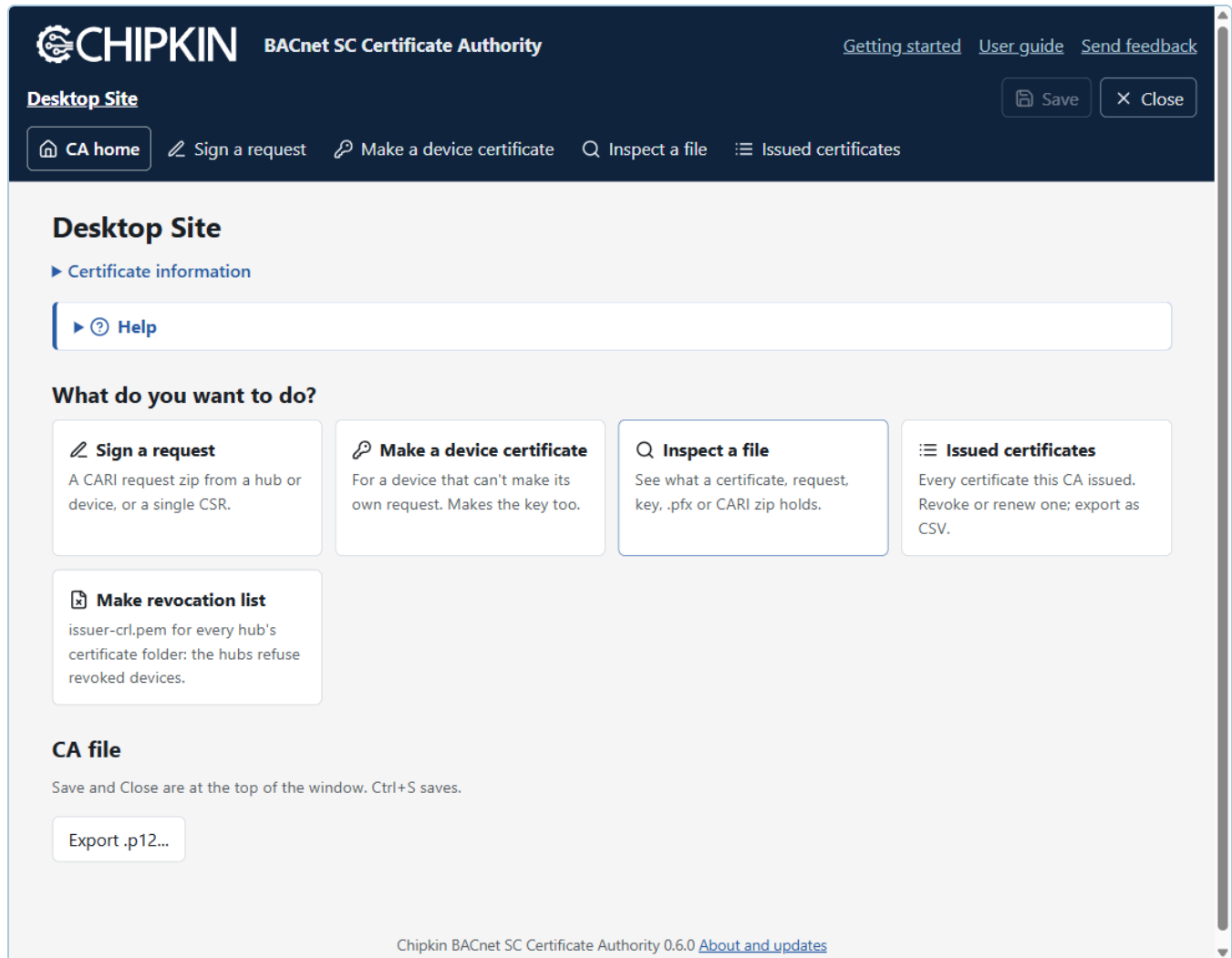


Figure 4-1. CA home, with Save and Close at the top of the window and one card for each task.

4.1 Saving and closing

The app keeps a **register**, which is the list of every certificate the CA has issued. The register is kept inside the CA file. When you sign, revoke, or renew a certificate, or make a revocation list, the change is only in the app until you save the CA file again.

- **Save** (or Ctrl+S) saves the CA file again. It is greyed out when there is nothing new to save.
- An **unsaved changes** badge beside the CA's name shows when there is something to save.
- **Close** asks first if there are unsaved changes. So does closing the app.

**IMPORTANT**

The CA closes itself after 15 minutes without use and clears its key from memory. Changes that weren't saved are lost. A warning appears 2 minutes before, with a **Keep it open** button. Save the CA file after every change.

4.2 Certificate information

On CA home, **Certificate information** shows the CA's details: name, key type, expiry, fingerprint, and the CA file. The fingerprint is a SHA-256 value that identifies one certificate.

4.3 The change log

The CA file records every change made to the CA, with the date: when it was created, each certificate issued or renewed, each one revoked, and each revocation list.

To save the log as a file, open **Certificate information** and click **Save the change log (.md)**. The file is called `<site>-changeLog.md`, and it includes changes not saved in the CA file yet. The log itself is stored inside the CA file.

5 Creating a site CA

Make one CA for each site, once.

Files out: the CA file (secret, password-protected) and a backup copy of it.



IMPORTANT

Save the backup copy before you sign anything, and keep the password in your company's password manager. The CA file and its password are the only way to sign, renew, or revoke certificates for this site.

1. On the start screen, click **Create a new site CA**.
2. Enter the **CA name (CN)** (for example "Example Building BACnet SC CA").
3. Optionally, enter the **Organization (O)**.
4. Leave **Valid for** at 10 years unless you have a reason to change it.
5. Leave the **Key type**, under **Advanced**, as **ECDSA P-256** unless a device on the site can only use RSA. Figure 5-1 shows the form.

CHIPKIN BACnet SC Certificate Authority [Getting started](#) [User guide](#) [Send feedback](#)

Create a new site CA

The CA signs every certificate on this site's BACnet/SC network. Make it once per site.

[▶ ⓘ Help](#)

CA name (CN)
 The name every certificate this CA signs shows as its issuer. Name it after the site, e.g. the building or campus. The CA file is named after it too.

Organization (O), optional
 The company that owns the site, shown in the CA's certificate. Leave it empty if you like: nothing checks it.

Valid for
 How long the CA can sign. When it expires every hub and device needs a certificate from a new CA, so make it long: 10 years suits most sites.

Password for the CA file
 Any length. Nobody can recover a lost password.
 It encrypts the CA's private key inside the CA file, and you need it every time you open the CA. It is not stored anywhere. **If it is lost, nobody can recover it, not even Chipkin:** the CA can't sign, renew or revoke anything again. Keep it in your company's password manager. With no password, anyone who gets the file can use the CA.

Password again
 The same password, to catch a typing mistake.

▼ **Advanced**

Key type
 The kind of key the CA signs with. Leave it at ECDSA P-256 unless a device on the site can only use RSA.

Chipkin BACnet SC Certificate Authority 0.6.0

Figure 5-1. Creating a site CA: a description beside each field, and the key type under Advanced.

1. Type a password in **Password for the CA file**. You need it every time you open the CA. Use a long one. Nobody can recover it, not even Chipkin.
2. Type the password again in **Password again**.
3. Click **Create CA**.
4. Click **Save CA file**. The CA file is called `<site>-ca.zip` (for example `Example-Building-BACnet-SC-ca.zip`).
5. Click **Save a second copy** and save it somewhere else, such as a USB stick kept in a safe place, or a network folder that is backed up.
6. Tick **I've saved a backup somewhere else**.
7. Tick **I've stored the password somewhere safe**.
8. Click **Continue**.

The fields of the **Create a new site CA** screen:

FIELD	FUNCTION	LEGAL VALUES
CA name (CN)	The name every certificate from this CA shows as its issuer. The CA file is named after it.	Text, up to 64 characters. Required.
Organization (O)	The company that owns the site, shown in the CA's certificate. Nothing checks it.	Text, up to 64 characters, or empty.
Valid for	How long the CA can sign. When it expires, every hub and device needs a certificate from a new CA.	10 years , 20 years, No expiry
Password for the CA file	Encrypts the CA's private key inside the CA file.	Any length. Empty only after a warning.
Password again	The same password, to catch a typing mistake.	The same as the password
Key type	The kind of key the CA signs with.	ECDSA P-256 , ECDSA P-384, RSA 3072-bit

* Bolded values are defaults.

ECDSA (Elliptic Curve Digital Signature Algorithm) and RSA (Rivest-Shamir-Adleman) are the two kinds of key the app can make.

5.1 Why the backup matters

The CA file holds the CA's private key. Without it, you can't sign a certificate for a new device, renew one that is about to expire, or revoke one. You would have to make a new CA and put a new certificate in the hubs and every device on the site.

Keep the CA file and its backup private. Anyone with the file and its password can make certificates your site will trust.

5.2 A CA file with no password

You can leave the password empty. The app first asks, in a dialog, whether to create the CA without a password. It warns that anyone who gets a copy of the CA file can then use the CA to sign certificates that every hub and device on the site trusts, or to revoke them. Click **Choose a password** to go back, or **Create without a password** to go on.

If you go on, the confirmation box in step 12 reads **I understand the CA file has no password**. To open the CA file, leave the password empty. CA home reminds you that the file has no password.



WARNING

A CA file with no password is protected only by where you keep it, and you can't add a password later (make a new CA instead). Do this only when the CA file is kept somewhere only you can reach.

6 Signing requests

Sign a request takes a request in, checks it, and saves a response zip. It handles a CARL request zip with one or many requests in it, and a single CSR file.

Two terms appear in every request:

- The **device instance** is the device's BACnet device instance number (0 to 4194302), as set on the device. It names the device's folder in the zip (`device-<instance>`).
- The **BACnet/SC port id** tells the app which of the device's BACnet/SC ports the certificate is for. It is usually 1. It names the port folder (`port-<id>`).

6.1 The Chipkin BACnet SC Hub's request

The hub makes its own private key and a request. The key never leaves the hub's computer. You sign the request and send back the response zip. Do this for the primary hub, and again for the failover hub if the site has one. Each hub has its own key and its own certificate.

- **File in:** `hub-cari-request.zip` (public, no password)
- **File out:** `hub-cari-request-response.zip` (public, no password)

IMPORTANT

Save the CA file again after you sign. Until you do, the new certificate is not in the register in your CA file.

1. On the hub's computer, get the hub's request, `hub-cari-request.zip`. Download it in the hub's set-up guide (step 2), or take it from the hub's certificate folder.
2. Copy `hub-cari-request.zip` to the computer with the app.
3. Open your CA and click **Sign a request**.
4. Choose `hub-cari-request.zip`.
5. Check the table (Figure 6-1). The hub's request shows as **Hub**, with the host names and addresses it asks for. They should be the names and addresses devices will use to reach the hub. If they are wrong, make the request again on the hub.

CHIPKIN BACnet SC Certificate Authority

Getting started User guide Send feedback

E2E Site BACnet/SC CA Save Close

CA home Sign a request Make a device certificate Inspect a file Issued certificates

Sign a request

Help

hub-cari-request: 3 requests

1 request has a problem and can't be signed.

Sign	Device / port	Name	Subject	Key	Hub or device	Problem
☒	device-12 / port-1	dev12	CN=Test device 12	ECDSA P-256	Device	None
☐	device-13 / port-1	weak	CN=weak rsa	RSA 1024-bit	Device	the request's key is RSA 1024-bit; BACnet/SC needs ECDSA P-256 or stronger, or RSA 2048-bit or stronger
☒	device-389999 / port-1	hub	CN=Chipkin BACnet SC Hub	ECDSA P-256	Hub names: dns:localhost	None

Certificates valid for

825 days (recommended)

Sign 2 and save the response Cancel

Chipkin BACnet SC Certificate Authority 0.6.0

Figure 6-1. Checking a request before signing. This example request has three entries, and one is refused, so the button reads Sign 2 and save the response.

1. Under **Certificates valid for**, choose how long the certificate is valid (825 days unless you have a reason to change it).
2. Click **Sign 1 and save the response**. The app saves `hub-cari-request-response.zip`.
3. Copy `hub-cari-request-response.zip` back to the hub's computer.
4. Install it in the hub's set-up guide (step 3, **Install the CA's response**). The hub starts using it at once.
5. Save the CA file again. Click **Save** at the top of the window, or press Ctrl+S. **Issued certificates** now lists the hub's certificate.

The hub's set-up guide is at `http://127.0.0.1:8080/setup` on the hub's computer. The hub's certificate folder is `C:\ProgramData\Chipkin\BACnetSCHub\certs` on Windows and `/etc/chipkin-bacnet-sc-hub/certs` on Linux. The hub's installers make the request the first time. To make the request or install the response zip from the command line, see the Chipkin BACnet SC Hub's user manual, "Making the request" and "Installing the CA's response".

NOTE

Two hubs on one site need different device instances and different device names. Both must be unique on the BACnet internetwork. See the Chipkin BACnet SC Hub's user manual.

The hub's request holds the CSR and never the hub's key:

```

hub-cari-request.zip
├── cert1/
│   ├── device-389022/
│   │   └── port-2/
│   │       ├── hub/           empty: marks a hub function
│   │       └── csr-hub.pem    the hub's request
│   └── request-notes.txt     a note to the CA

```

389022 is the hub's default device instance. The response zip is the same tree with the hub's certificate and the CA's certificate added:

```

hub-cari-request-response.zip
├── cert1/
│   ├── device-389022/
│   │   ├── port-2/
│   │   │   ├── hub/
│   │   │   ├── csr-hub.pem
│   │   │   └── opr-hub.pem    added: the hub's certificate
│   ├── issuer/
│   │   └── iss-1.pem         added: the CA's certificate
│   ├── request-notes.txt
│   └── response-notes.txt    added: a note from the app

```

6.2 Devices that make their own request

Most BACnet/SC devices and tools make their own key and give you a request (the key stays on the device). The request comes in one of two forms.

A CARI request zip, from the device's own tool, any CARI-compatible tool, or another hub. Sign it exactly as for the hub. Choose the zip in **Sign a request**, check the table, and sign. Then follow steps 11 to 13 of the next procedure.

A single request file (a `.csr`, `.pem`, `.der`, `.req`, or `.p10` file). A single request doesn't tell the app which device or port it belongs to, so you enter that:

1. Open your CA and click **Sign a request**.
2. Choose the request file.
3. Enter the **Device instance**.
4. Check the **BACnet/SC port id**.
5. Check the **Name**, which the app fills in for you.
6. Under **This port is**, choose **A device (client)** or **A hub (server and client)**.
7. Click **Continue**.
8. Check the table.

9. Under **Certificates valid for**, choose how long the certificate is valid.
10. Click **Sign 1 and save the response**. The app saves `<request file name>-response.zip`.
11. Give the response zip to the device's tool to install. If the tool can't take the zip, take out `opr-<name>.pem` (the device's certificate, its Operational Certificate) and `cert1/issuer/iss-1.pem` (the CA's certificate, its Issuer Certificate). A device with two places (slots) for issuer certificates gets `iss-1.pem` in both.
12. Set the device's primary hub URI (uniform resource identifier, the hub's address) to the primary hub, for example `wss://192.168.1.20:4443/`. Set its failover hub URI to the failover hub, if the site has one. A hub URI starts with `wss://`, which stands for WebSocket Secure.
13. Save the CA file again.

The fields of the **Sign a request** screen (the first four appear only for a single request file):

FIELD	FUNCTION	LEGAL VALUES
Device instance	The device the request is for.	A whole number from 0 to 4194302
BACnet/SC port id	The device's BACnet/SC port the certificate is for.	Text. 1
Name	The short name in <code>csr-<name>.pem</code> and <code>opr-<name>.pem</code> .	Text. The request's file name
This port is	Whether the certificate is for a device or a hub.	A device (client) , A hub (server and client)
Certificates valid for	How long each certificate signed from this request is valid.	825 days , 1 year, 2 years, 5 years, No expiry

The response zip is public and has no password, unless the request carried `key-<name>.pem` files. In that case the response zip holds those keys and is secret.



WARNING

Some tools put the device's private key (`key-<name>.pem`) in the request zip. The app ignores the key but keeps it, so the response zip then holds the key too, unencrypted. Treat both zips as secret.

A device's request zip follows the same layout as the hub's. A typical one:

```

device-cari-request.zip
├── cert1/
│   ├── device-1234/           the device instance
│   │   └── port-1/           one folder per BACnet/SC port
│   │       ├── csr-ahu7.pem  the device's request
│   │       └── key-ahu7.pem  optional: the device's private key
│   ├── request-notes.txt     optional: a note to the CA
│   └── vendor-data           optional: copied into the response unchanged

```

6.3 What the app checks

The app checks every request in the file and shows each one in the table, with any problem, before you choose which to sign. A request that fails is refused, and the others can still be signed. A request is refused when any of these is true:

- It isn't a certificate signing request.

- Its key is weaker than ECDSA P-256 or RSA 2048-bit, or of a type the app can't sign. The app signs ECDSA keys on P-256, P-384, or P-521, and RSA keys of 2048 bits or more.
- Its signature doesn't verify.
- Its subject (the name in the request) is empty.
- A `key-<name>.pem` beside it is not the key the request is for.

When a request is not signed, the response zip holds `cert1/errors.txt`, with one line for each. That includes requests you cleared in the table, which are listed as "not selected for signing".

A hub's certificate can be used at both ends of a connection (as a server and as a client), and keeps the host names or addresses the request asks for. A device's certificate is for a client only.

A CARL zip may hold only the names the format allows. An extra file such as a `readme.txt`, a `.cer` file, or a `__MACOSX/` folder makes the zip invalid. The app refuses it and names the file.

Device vendors who make requests with their own tools can get the format details from Chipkin support.

7 Making a device's key and certificate

Some devices and tools can't make their own key. For these, the app makes the key and the certificate together, with **Make a device certificate**. Use **Sign a request** instead whenever the device can make its own request.

Files out:

- `device-<instance>-<name>-response-PRIVATE.zip` (secret, no password)
- `device-<instance>-<name>-PRIVATE.pfx` (secret, empty password), if you tick its box
- `iss-1.cer` (public), if you tick its box

A `.pfx` file is a PKCS#12 file: one file that holds a certificate with its private key. `iss-1.cer` is the CA's certificate in DER, the binary form that Windows tools read.



WARNING

The response zip and the `.pfx` hold the device's private key, not encrypted. Neither has a password. A CARL zip has no place for one, and the `.pfx` password is empty. Protect the files themselves. Carry them privately (not by email), install them, then delete the copies.

1. Open your CA and click **Make a device certificate**.
2. Enter the **Device instance**, as set on the device.
3. Enter the **BACnet/SC port id**, usually 1.
4. Enter a short **Name** for the files (for example `ahu-7`).
5. Optionally, enter a **Certificate name (CN)** and an **Organization (O)**.
6. Under **This port is**, choose **A device (client)**, or **A hub (server and client)** for a hub.
7. For a hub, optionally enter the **Hub's host names or IP addresses**.
8. Under **Valid for**, choose how long the certificate is valid.
9. For YABE (Yet Another BACnet Explorer, a third-party BACnet tool) or another Windows tool, open **Advanced** and tick both boxes under **Also make, for Windows tools such as YABE** (Figure 7-1).

CHIPKIN

BACnet SC Certificate Authority

[Getting started](#) [User guide](#) [Send feedback](#)

Review CA

Save

Close

CA home

Sign a request

Make a device certificate

Inspect a file

Issued certificates

Make a device certificate

For a device that can't make its own request. This makes the device's key and certificate.

Help

Device instance

The device's BACnet device instance number (0 to 4194302), as set on the device. It names the folder in the response zip (device-<instance>).

BACnet/SC port id

1

Which of the device's BACnet/SC ports this certificate is for, usually 1. It names the port folder (port-<id>).

Name (for the file names, e.g. ahu-7)

A short name for this device: the files are opr-<name>.pem and key-<name>.pem, and the zip is named after it.

Certificate name (CN), optional

Same as the name

The name inside the certificate, shown by tools that inspect it. Empty: the name above.

Organization (O), optional

The company that owns the device, shown in the certificate. Nothing checks it.

This port is

☒ A device (client)

☐ A hub (server and client)

A device connects to the hub. A hub also accepts connections, so its certificate can be used as a server certificate too (the response gets the hub/ folder).

Valid for

825 days (recommended)

How long the certificate works. Renew it from Issued certificates before it expires; the app warns 60 days ahead.

Advanced

Key type

ECDSA P-256 (recommended)

The kind of key made for the device. Leave it at ECDSA P-256 unless the device can only use RSA.

Also make, for Windows tools such as YABE

☐ A .pfx file (certificate, key and issuer; no password)

☐ iss-1.cer (the CA certificate)

Only for tools that can't read the response zip. The .pfx holds the device's private key with an empty password; protect the file itself.

Make key and certificate

Chipkin BACnet SC Certificate Authority 0.6.0

Figure 7-1. Make a device certificate, with Advanced open: the key type and the extra files for Windows tools such as YABE.

1. Click **Make key and certificate**.
2. Click **Save response zip (private)**. If you ticked the boxes in step 9, also click **Save .pfx (private)** and **Save iss-1.cer**.
3. Install the files on the device (see "Installing on the device" in this section).
4. Delete the copies of the zip and the `.pfx` once the device is set up.
5. Save the CA file again.

The fields of the **Make a device certificate** screen (the last three are under **Advanced**):

FIELD	FUNCTION	LEGAL VALUES
Device instance	The device's BACnet device instance number, as set on the device.	A whole number from 0 to 4194302
BACnet/SC port id	The device's BACnet/SC port the certificate is for.	Text. 1
Name	A short name for the device, used in the file names.	Text, for example <code>ahu-7</code>

FIELD	FUNCTION	LEGAL VALUES
Certificate name (CN)	The name inside the certificate, shown by tools that inspect it.	Text. The same as the name
Organization (O)	The company that owns the device. Nothing checks it.	Text, or empty
This port is	Whether the certificate is for a device or a hub.	A device (client) , A hub (server and client)
Hub's host names or IP addresses	For a hub only. The names and addresses devices use to reach the hub.	Names and addresses separated by commas, or empty
Valid for	How long the certificate is valid.	825 days , 1 year, 2 years, 5 years, No expiry
Key type	The kind of key made for the device. Change it only if the device can only use RSA.	ECDSA P-256 , ECDSA P-384, RSA 2048-bit, RSA 3072-bit
A .pfx file	Also makes the <code>.pfx</code> for Windows tools such as YABE.	Ticked, cleared
iss-1.cer	Also makes the CA's certificate as <code>iss-1.cer</code> .	Ticked, cleared

The response zip holds the device's key beside its request and certificate:

```

device-<instance>-<name>-response-PRIVATE.zip
├─ cert1/
│   └─ device-<instance>/
│       └─ port-<id>/
│           └─ hub/
│               └─ csr-<name>.pem    only for a hub port
│                   the request the app made
│               └─ key-<name>.pem    the device's private key, NOT encrypted
│                   the device's certificate
│               └─ opr-<name>.pem
│   └─ issuer/
│       └─ iss-1.pem                the CA's certificate
└─ response-notes.txt               lists each file and where it goes

```

7.1 Installing on the device

The **Chipkin BACnet Explorer** imports the response zip directly. Set its hub URI to the hub computer's LAN address, for example `wss://192.168.1.20:4443/`, not `127.0.0.1` or `localhost`, even when the Explorer runs on the hub's own computer.

YABE needs the `.pfx` file and `iss-1.cer`. The `.pfx` is secret and has no password. See the warning at the start of this section. When Windows or YABE asks for a password, leave it blank.

WHAT	NEEDS
The <code>.pfx</code> file	Windows 10 version 1709 or later, Windows 11, or Windows Server 2019 or later. Windows 7, Windows 8.1, and Windows Server 2016 can't open it. On those, use the certificate and key files from the zip.
TLS 1.3, which BACnet/SC requires	Windows 11 or Windows Server 2022. Windows has TLS 1.3 only from those versions.
The Chipkin BACnet SC Hub	Started with <code>--sc-accept-device-without-hello on</code> .

BACnet/SC connection messages carry a Hello option, and YABE leaves it out, so the hub refuses YABE unless that setting is on. The Chipkin BACnet SC Hub's user manual, "BACnet/SC compatibility", explains how to set it.

Any other tool uses `opr-<name>.pem`, `key-<name>.pem`, and `cert1/issuer/iss-1.pem` from the zip.

8 Revoking and renewing

8.1 Revoking a certificate and making the revocation list

Revoke a certificate when a device is lost, stolen, replaced, or taken out of service, or when its key may be known to someone else. A revoked device can't connect to a hub once that hub has the new revocation list. The list is a certificate revocation list (CRL). It holds the serial numbers of the revoked certificates, signed by the CA. A serial number is the unique number the CA gives each certificate.

File out: `issuer-crl.pem` (public, no password), for every hub.



CAUTION

Revoking can't be undone. A revoked certificate stays revoked, and the device needs a new certificate from a new request.



CAUTION

Once a hub has a revocation list, someone must replace it before its date, every time, or the hub refuses every device.

1. Open **Issued certificates**.
2. Find the certificate and click **Revoke...**
3. Choose the **Reason** (Figure 8-1).

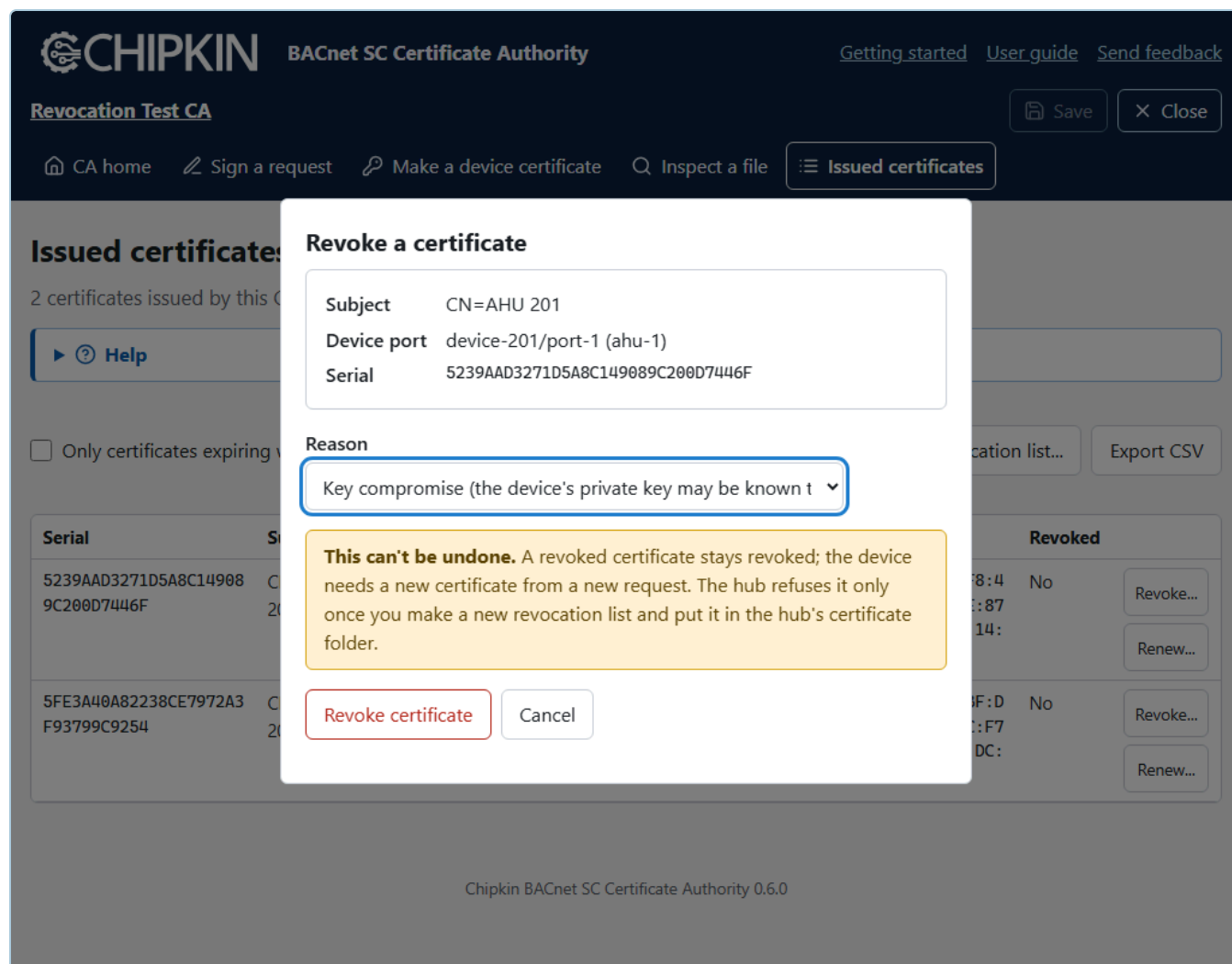


Figure 8-1. Revoking a certificate: the certificate, the reason, and the warning that it can't be undone.

1. Click **Revoke certificate**.
2. Click **Make revocation list....**
3. Leave **Valid for** at 30 days unless you have a reason to change it (Figure 8-2).

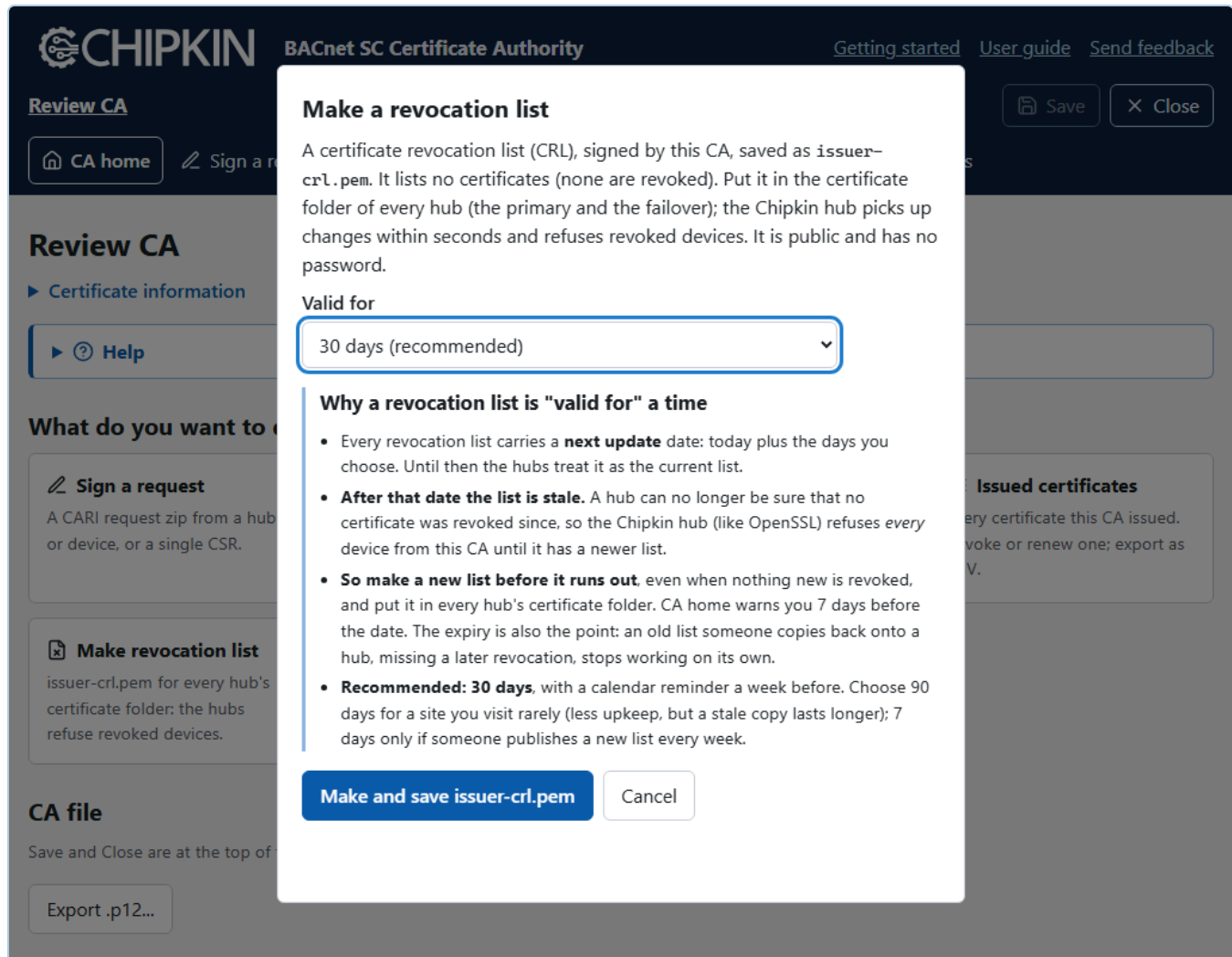


Figure 8-2. Making a revocation list: what "valid for" means.

1. Click **Make and save issuer-crl.pem**. The file is always called `issuer-crl.pem`.
2. Put `issuer-crl.pem` in the certificate folder of every hub (the folder that holds `hub-cari-request.zip`).
3. Save the CA file again.

The reasons are **Unspecified**, **Key compromise**, **CA compromise**, **Affiliation changed**, **Superseded**, and **Cessation of operation**.

The **Make revocation list** card on CA home opens the same dialog as step 5.

The Chipkin BACnet SC Hub notices the new file within a few seconds and restarts BACnet/SC. Every device reconnects briefly, and the revoked device is refused. You don't need to restart the hub.

8.2 Keeping the revocation list current

Every revocation list carries a "next update" date, which is the day you make it plus the days you choose. After that date the list is stale, and the Chipkin BACnet SC Hub (like OpenSSL) refuses every device from this CA until it has a newer list.

The date also protects you. An old list that is copied back onto a hub stops working when its date passes. Make a new list before the date, even when nothing new is revoked.

CA home shows when the next list is due, and warns 7 days before. Each new list holds every certificate revoked so far.

The **Make a revocation list** dialog has one field:

FIELD	FUNCTION	LEGAL VALUES
Valid for	The days from today until the list's "next update" date.	7 days, 30 days , 90 days

Use 30 days with a calendar reminder a week before. Choose 90 days for a site you visit rarely (less upkeep, but a stale copy lasts longer). Choose 7 days only if someone publishes a new list every week.

8.3 Renewing a certificate

Certificates expire 825 days after signing, unless you chose otherwise. CA home and **Issued certificates** warn 60 days before.

1. Open **Issued certificates**.
2. Click **Renew...** on the certificate.
3. Under **New certificate valid for**, choose how long the new certificate is valid.
4. Leave **Also revoke the old certificate (superseded)** cleared until the device has the new certificate.
5. Click **Renew and save the response**. The app saves `device-<instance>-<name>-renewed-response.zip`. If you cancel the save dialog, the renewal is dropped.
6. Install the response zip on the device the same way as the first time.
7. Save the CA file again.

The fields of the **Renew a certificate** dialog:

FIELD	FUNCTION	LEGAL VALUES
New certificate valid for	How long the new certificate is valid.	825 days , 1 year, 2 years, 5 years, No expiry
Also revoke the old certificate (superseded)	Revokes the old certificate as part of the renewal.	Ticked, cleared

The renewal uses the same request and the same key, so nothing changes on the device except the certificate. The response zip is public and has no password.

Once the device is using the new certificate, you can revoke the old one as **Superseded** and make a new revocation list.

You can also renew by signing a new request from the device.

8.4 Renewing a hub's certificate

The Chipkin BACnet SC Hub's user manual, "Renewing the hub's certificate", describes the hub's side. From the app's side there are two cases:

- **The hub keeps its key.** Click **Renew...** on the hub's certificate in **Issued certificates**. The app signs the same request again. Install the response zip on the hub as before.
- **The hub makes a new key.** The hub writes a new `hub-cari-request.zip`. Sign it with **Sign a request**, as in section 6, "The Chipkin BACnet SC Hub's request".

9 The register

Issued certificates lists every certificate the CA has signed: serial number, name, device and port, hub or device, when it was issued and when it expires, and whether it is revoked. Certificates that expire within 60 days are marked. Figure 9-1 shows a register with a revoked certificate and a renewed one.

CHIPKIN BACnet SC Certificate Authority [Getting started](#) [User guide](#) [Send feedback](#)

Revocation Test CA unsaved changes Save Close

[CA home](#) [Sign a request](#) [Make a device certificate](#) [Inspect a file](#) **Issued certificates**

Issued certificates

3 certificates issued by this CA. 1 revoked.

[Help](#)

Saved device-202-ahu-2-renewed-response.zip: certificate 664C4491E96F4E1D19A0782A9F872B04 renews 5FE3A40A82238CE7972A3F93799C9254, valid until 2027-10-08. Save the CA file again to keep it in the register.

☐ Only certificates expiring within 60 days Make revocation list... Export CSV

Last revocation list: number 1, next update due 2026-11-07.

Serial	Subject	Folder	Issued	Expires	SHA-256 fingerprint	Revoked
5239AAD3271D5A8C1490 89C200D7446F	CN=AHU 201	device-201/port-1 ahu-1 (device)	2026-10-08	2029-01-10	AF:51:D6:BE:3C:9F:80:F 8:40:76:5D:BD:EF:55:D1 :DE:87:8F:17:3D:D3:59: 34:59:14:CE:DB:86:83:0 C:22:FC	revoked Key compromise, 2026-10-08
5FE3A40A82238CE7972A 3F93799C9254 renewed by 664C4491E96F4E1D19A0 782A9F872B04	CN=AHU 202	device-202/port-1 ahu-2 (device)	2026-10-08	2029-01-10	EF:23:A2:EA:96:98:A9:B F:D4:AD:17:1A:05:32:D4 :0C:F7:FE:F2:DB:1B:3B: A7:02:DC:0A:44:70:48:3 5:31:14	No Revoke... Renew...
664C4491E96F4E1D19A0 782A9F872B04 renews 5FE3A40A82238CE7972A 3F93799C9254	CN=AHU 202	device-202/port-1 ahu-2 (device)	2026-10-08	2027-10-08	D0:65:34:8E:6C:A0:41:8 5:9D:E1:34:A4:86:97:BA :23:EF:F6:5E:97:CF:48: 86:85:91:EF:AF:CB:97:5 5:63:8F	No Revoke... Renew...

Chipkin BACnet SC Certificate Authority 0.6.0

Figure 9-1. Issued certificates: one certificate revoked, and one renewed, with Revoke and Renew beside each current certificate.

Export CSV saves the list as a comma-separated values (CSV) file, `<site>-register.csv`, for a spreadsheet. It holds no keys and has no password.

The register is kept inside the CA file. That is why the app asks you to save the CA file again after you sign, revoke, or renew. See section 4, "Saving and closing".

10 Files and passwords

10.1 Which files have a password

Only two files have a password: the **CA file** and a `.p12` export of the CA. Every other file has none, and needs none. The files that hold a device's private key without a password are secret, so protect the files themselves.

FILE	PASSWORD?	WHO CHOOSES IT	IF YOU LOSE IT
The CA file, <code><site>-ca.zip</code>	Yes, every time you open the CA	You, when you create the CA	Nobody can recover it
The CA exported as <code>.p12</code>	Yes, to open the <code>.p12</code>	You, when you export it	Export a new <code>.p12</code> from the CA file
A device's <code>.pfx</code> (for YABE)	No. The password is empty	Nobody	Nothing to lose
<code>device-<instance>-<name>-response-PRIVATE.zip</code>	No. The key in it is not encrypted	Nobody	Nothing to lose
Requests, response zips from Sign a request , <code>issuer-crl.pem</code> , <code>iss-1.cer</code> , the register CSV	No	Nobody	Nothing to lose

The CA file's password encrypts the CA's private key inside the file. Any length is accepted (for a CA file with no password, see section 5, "A CA file with no password"). Keep the password in a password manager.



WARNING

Nobody can recover a lost CA file password, not even Chipkin. Without it the CA can't sign, renew, or revoke anything again. You would make a new CA and give the hubs and every device new certificates.

The Chipkin BACnet SC Hub can protect its own key with a password, which is set on the hub. See the Chipkin BACnet SC Hub's user manual, "Password-protected private keys".

10.2 Every file and where it goes

FILE	WHAT IS INSIDE	SECRET?	GOES TO
<code><site>-ca.zip</code> , the CA file	The CA's certificate, its encrypted private key, the register, and the change log	Secret	Stays with you, with a backup copy somewhere else
<code>hub-cari-request.zip</code> , the hub's request	The hub's CSR (no private key)	Public	From the hub to the app
A CARL request zip	One or more CSRs, one folder per device and port	Public, unless it carries <code>key-<name>.pem</code> files	From the device's tool to the app
<code><name>.csr</code> , <code>.pem</code> , <code>.der</code> , <code>.req</code> , or <code>.p10</code>	One CSR	Public	To the app
<code><request>-response.zip</code> , the response zip	The request, each certificate, and the CA's certificate	Public, unless the request carried <code>key-<name>.pem</code> files	Back to the hub or device

FILE	WHAT IS INSIDE	SECRET?	GOES TO
<code>device-<instance>-<name>-response-PRIVATE.zip</code>	A device's private key (not encrypted), its certificate, and the CA's certificate	Secret	To the device
<code>device-<instance>-<name>-PRIVATE.pfx</code>	The same, as PKCS#12, with an empty password	Secret	To the computer running YABE
<code>iss-1.cer</code>	The CA's certificate, in DER form	Public	To Windows tools such as YABE
<code>issuer-crl.pem</code> , the revocation list	The revoked serial numbers, signed by the CA	Public	Into the certificate folder of every hub
<code><site>-register.csv</code>	The register as a spreadsheet	Public (your site inventory)	Your records
<code><site>-ca-PRIVATE.p12</code>	The CA's certificate and private key	Secret	Only to OpenSSL or another CA tool, when moving the CA

**WARNING**

Never give the CA file to a hub, a device, or a device vendor. They get a response zip.

PEM is the text form of a certificate, request, or key, DER is the binary form, and PKCS#12 (`.pfx` or `.p12`) is one file that holds a certificate with its private key. Appendix B, "Glossary of terms", has every term.

10.3 The CA file

The CA file is `<site>-ca.zip`, named from the CA's name (for example `Example-Building-BACnet-SC-ca.zip`). It is in the CARI format, laid out like a response zip:

```

<site>-ca.zip
├─ cert1/
│   ├─ device-<instance>/
│   │   └─ router/
│   │       └─ port-<id>/
│   │           └─ hub/
│   │               └─ csr-<name>.pem
│   │               └─ opr-<name>.pem
│   └─ issuer/
│       └─ iss-1.pem
├─ response-notes.txt
└─ vendor-data

```

one folder per device with a certificate
only for a device that routes between networks
one folder per BACnet/SC port
only for a hub port
the request (public)
the certificate (public)
the CA's certificate (public)
a note: what this file is, and that it is secret
the CA's private key, ENCRYPTED with your password, the register, and the change log

The device folders hold the newest certificate that isn't revoked for each device port.

11 Checking any file

Inspect a file (on the start screen, or at the top of the window once a CA is open) shows what a file is and checks it. It reads these files:

- A certificate
- A request
- A key
- A `.pfx`
- A CARI zip
- A CA file
- A revocation list (`issuer-crl.pem`)

With your CA open, it also tells you whether your CA signed the file, and for a revocation list, whether it is out of date and agrees with the register (Figure 11-1). You don't need a CA open to use it.

CHIPKIN BACnet SC Certificate Authority

[Getting started](#) [User guide](#) [Send feedback](#)

Revocation Test CA unsaved changes Save Close

[CA home](#) [Sign a request](#) [Make a device certificate](#) [Inspect a file](#) [Issued certificates](#)

Inspect a file

A certificate, request (CSR), key, .pfx/.p12, CARI zip, CA file or revocation list (issuer-crl.pem): what it is, every field, and checks against the open CA.

[Help](#)

File to inspect

Choose File No file chosen

or drop the file here

issuer-crl.pem: Revocation list (CRL) (PEM)

Revocation list (CRL)

- OK** current: the next one is due 2026-11-07 16:33:53 UTC
- OK** signed by the open CA (O=Chipkin Test, CN=Revocation Test CA)
- OK** the serial it lists is revoked in the register

Issuer	O=Chipkin Test, CN=Revocation Test CA
CRL number	1

Figure 11-1. Inspecting a revocation list with the CA open: current, signed by the open CA, and in step with the register.

A `.pfx` made by the app opens without a password. Only an encrypted key, a CA `.p12`, or a `.pfx` from elsewhere asks for one.

Inspect a file only reads. Nothing is changed or saved.

12 Moving to a new computer and backing up

12.1 Moving to a new computer

1. Install the app on the new computer.
2. Copy the CA file (`<site>-ca.zip`) to it.
3. Open the CA file from the start screen, with the same password.

The register is inside the CA file, so nothing else has to move. The list of recent CA files on the start screen is only a list of where the files are.

12.2 Backing up

The CA file is the only thing to back up. Each time you sign, revoke, renew, or make a revocation list, the CA file changes when you save it. Copy the saved file over your backup afterwards. Keep the backup somewhere else, such as a USB stick kept in a safe place, or a network folder that is backed up. Keep it as private as the CA file itself.

The password is not in the CA file or in its backup. Keep it in a password manager.

12.3 If the CA file or its password is lost

Nobody can recover a lost CA file or a lost password, not even Chipkin. With either one lost:

- Nothing new can be signed, so a new or replaced device can't get a certificate from this CA.
- No certificate can be renewed or revoked.
- No new revocation list can be made. Hubs that have a revocation list refuse every device once its date passes.

The site then needs a new CA, and a new certificate from it for every hub and every device. If a backup copy of the CA file exists, open that copy with the same password instead.

12.4 Moving the CA to another tool

Export .p12... saves the CA's certificate and private key as a `.p12` file, for OpenSSL or another CA tool. The `.p12` is secret. Give it only to the tool you are moving the CA to.

1. On CA home, click **Export .p12....**
2. In **Password for the .p12 file (required)**, type a password of at least 12 characters. It doesn't have to be the CA file's password.
3. Type it again in **Password again**.
4. Tick **I understand this file holds the CA's private key**.
5. Click **Export .p12**. The app saves `<site>-ca-PRIVATE.p12`.

If you lose this password, that `.p12` can't be opened. Your CA file is not affected, so you can export again.

13 Privacy: what is sent and what is never sent

13.1 What is sent, and when

The app connects to Chipkin's update service (`app-updates.chipkin.com`) for the update check and for feedback. A download that you start comes from the update service or another chipkin.com address. All of these use HTTPS (Hypertext Transfer Protocol Secure). The app makes no other connections.

WHAT	WHEN
Update check	When the app starts (after a few seconds), every 24 hours while it is open, and when you click Check for updates
Download	Only when you click Download and install
Feedback	Only when you click Send feedback on the Send feedback page

Update check. The app asks for the newest version number of the app for your kind of system. It sends the app's name, `win32` or `linux` , the processor type (`x64`), and a random number. Like any web request, it also shows your computer's internet address to the service. Nothing else is sent. To turn the automatic check off, open **About and updates** and clear **Check for updates automatically**.

Download. The app sends a request for the installer file.

Feedback. The app sends your message, your email address if you give one, and the app's name and version (the service files feedback by them). With **Include diagnostics** ticked, it also sends the app's version and the operating system (name, version, and processor type), which the page shows before you send. Figure 13-1 shows the page.

CHIPKIN BACnet SC Certificate Authority

[Getting started](#) [User guide](#) [Send feedback](#)

[Back](#)

Send feedback

Tell Chipkin what you think of the app, or what went wrong.

Your message

The revocation list step is clear. Thanks!

What works, what doesn't, what you'd like. Don't paste passwords, keys or certificates.

Your email (optional)

tester@example.coma

Only if you'd like an answer.

☒ Include diagnostics: the app's version and the operating system

App version: 0.6.0
Operating system: Windows_NT 10.0.26300 (win32, x64)

Sent to Chipkin's feedback service, over HTTPS: your message, the email if you give one, the diagnostics if ticked, and the app's name and version. Never your CA, keys, file names or register.

Send feedback

Figure 13-1. The Send feedback page: the message, an optional email address, and the diagnostics shown before sending.

If there is no network, the automatic check fails quietly and tries again later, and **Check for updates** reports that it couldn't reach the service. The app works the same with no network at all.

13.2 What is never sent anywhere

- None of the connections above carries anything from your CA.
- Your CA file, its password, private keys, requests, certificates, and the register stay on your computer, in the files you choose to save.
- There is no account, no usage tracking, and no crash reporting. Feedback holds only what you type (and the diagnostics, if you tick the box). It never holds your CA, keys, file names, or register.
- The password is never stored. The list of recent CA files on the start screen holds their names and folders only.

A device's response zip from **Make a device certificate**, a device's `.pfx`, and a `.p12` export of the CA have `PRIVATE` in their file names. The CA file is secret too. Keep all of them private.

14 Routine upkeep

TASK	WHEN
Replace the revocation list on every hub	Before its "next update" date, every time, once a hub has a list. CA home warns 7 days before.
Renew certificates	Before they expire. CA home and Issued certificates warn 60 days before, when the CA is open.
Save the CA file, then copy it over the backup	After every change: signing, revoking, renewing, or making a revocation list.
Keep the CA file's password in a password manager	Always. Nobody can recover it.

The app shows its warnings only while the CA is open, so put the dates in a calendar too. See section 8, "Revoking and renewing".

15 Troubleshooting

15.1 Installing and opening

For "Windows protected your PC" and an ApplImage that doesn't start, see the notes in section 2, "Installing".

SYMPTOM	CAUSE	WHAT TO DO
Double-clicking the CA file opens a zip tool	A CA file is a <code>.zip</code> file, which belongs to your zip tool.	Open it from the start screen with Choose CA file... or Recent CA files .
The app reports that the file is a request or a response, not a CA file	You chose a request zip or a response zip when opening a CA.	Choose <code><site>-ca.zip</code> . Requests and response zips are signed or inspected, not opened.
The CA file's password is lost	Nobody can recover it, not even Chipkin.	See section 12, "If the CA file or its password is lost".
The CA closed while you were away	The CA closes itself after 15 minutes without use, and unsaved changes are lost.	Open the CA file again. Save after every change, or click Keep it open when the warning appears.
Check for updates couldn't reach the service	The computer has no network connection to Chipkin's update service.	Nothing is needed. The app works the same with no network.

15.2 Signing

SYMPTOM	CAUSE	WHAT TO DO
"this is the CA file, not a request"	You chose the CA file in Sign a request .	Choose the request zip or the CSR file.
A request in the table has a problem and can't be signed	The request failed a check (see section 6, "What the app checks").	Have the device or its tool make a new request that meets the rules.
The app refuses a request zip and names a file	The zip holds a file or folder the CARI format doesn't allow.	Ask the device's tool vendor for a clean request.
Save is greyed out	There is nothing new to save.	Nothing is needed.

15.3 Hubs and devices

SYMPTOM	CAUSE	WHAT TO DO
A hub refuses every device	The hub's <code>issuer-crl.pem</code> is past its "next update" date.	Make a new revocation list and put it in every hub's certificate folder.
A revoked device still connects	The hub doesn't have the new revocation list yet.	Put the new <code>issuer-crl.pem</code> in the certificate folder of every hub.
Windows or YABE asks for the <code>.pfx</code> password	The <code>.pfx</code> has an empty password.	Leave the password blank.
Windows can't open the <code>.pfx</code>	The <code>.pfx</code> needs Windows 10 version 1709, Windows Server 2019, or later.	On Windows 7, Windows 8.1, and Windows Server 2016, use the certificate and key files from the zip.

SYMPTOM	CAUSE	WHAT TO DO
YABE can't connect to the hub	YABE needs TLS 1.3 (Windows 11 or Windows Server 2022), and the hub must accept a device without the Hello option.	Run YABE on a supported Windows version, and start the hub with <code>--sc-accept-device-without-hello on</code> .
The Chipkin BACnet Explorer can't reach the hub at <code>127.0.0.1</code>	The Explorer needs the hub computer's LAN address.	Set the hub URI to the LAN address, for example <code>wss://192.168.1.20:4443/</code> .

16 Version and licensing

- **Version.** This manual describes version 0.6.1 of the Chipkin BACnet SC Certificate Authority (part number CAS-1000-22). **About and updates** (at the bottom of the window) shows the version you are running.
- **Licensing.** The Chipkin BACnet SC Certificate Authority is software of Chipkin Automation Systems Inc. Licensing is arranged with Chipkin. Use <https://store.chipkin.com/contact-us> or email sales@chipkin.com.
- **Related products.** The Chipkin BACnet SC Hub (part number CAS-1000-21) is the hub whose requests the app signs. The Chipkin BACnet Explorer imports a device's response zip.

17 Contact us for support

Chipkin provides technical support for the Chipkin BACnet SC Certificate Authority.

- **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 app's version, from **About and updates**
- The operating system
- What you did and what you saw



WARNING

Never send the CA file, its password, or a private key to anyone, including Chipkin support.

You can also use **Send feedback** at the top of the window.

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

A Document history

This table lists the released versions of this manual.

DOCUMENT REVISION	DATE	PRODUCT VERSION	CHANGES
1.0	2026-Oct-15	0.6.1	First release.

B Glossary of terms

TERM	MEANING
BACnet Secure Connect (BACnet/SC)	The BACnet data link that connects every device to a hub over TLS 1.3, with a certificate at each end.
CA (certificate authority)	The signing identity that signs the certificates of every hub and device on a site.
CA file	The file <code><site>-ca.zip</code> , which holds the CA's certificate, its encrypted private key, the register, and the change log.
CARI	Certificate Authority Requirements Interchange, the folder layout BACnet uses to take requests to a CA and bring certificates back.
Certificate	A file, signed by the CA, that a hub or device presents to prove who it is.
CRL (certificate revocation list), revocation list	The list of revoked certificates' serial numbers, signed by the CA and saved as <code>issuer-crl.pem</code> .
CSR (certificate signing request)	A request: the public key and name of a hub or device, sent to the CA for signing.
DER	Distinguished Encoding Rules, the binary form of a certificate, request, or key.
Device instance	The device's BACnet device instance number (0 to 4194302), as set on the device.
ECDSA	Elliptic Curve Digital Signature Algorithm, one of the two kinds of key the app can make.
Failover hub	A second hub that devices connect to when the primary hub is down.
Hub	The BACnet/SC device that every other device on the network connects to.
Issuer certificate	The CA's own certificate (<code>iss-1.pem</code>), which every hub and device loads so that it trusts certificates the CA signed.
PEM	Privacy-Enhanced Mail, the text form of a certificate, request, or key.
PKCS#12 (<code>.p12</code> , <code>.pfx</code>)	Public-Key Cryptography Standards number 12, one file that holds a certificate with its private key.
Port id	Which of a device's BACnet/SC ports a certificate is for, usually 1.
Private key	The secret half of a key pair, which a hub or device (or the CA) keeps to itself.
Register	The list of every certificate the CA has issued, kept inside the CA file and shown in Issued certificates .
Renew	To sign the same request again, giving the device a new certificate with a new expiry date for the same key.
Request	A CSR, either as a single file or inside a CARI request zip.
Response zip	The zip file the app saves after signing, which holds the request, the certificate, and the CA's certificate.
Revoke	To mark a certificate as no longer trusted, which can't be undone.
RSA	Rivest-Shamir-Adleman, one of the two kinds of key the app can make.
Serial number	The unique number the CA gives each certificate.
Site CA	The one CA you make for a site.
TLS	Transport Layer Security, the encryption BACnet/SC connections use.