



BridgeLink Quick Start Guide macOS

For New Installations of BridgeLink 26.9.0

Table of Contents

1. About This Guide3

2. Getting the Software5

3. Installing BridgeLink6

4. Choosing and Connecting the Database8

5. The Service Account10

6. Starting BridgeLink for the First Time16

7. Installing WebAdmin18

8. Finding the Server ID22

9. Replacing the Certificates24

10. The Data Pruner28

11. Troubleshooting31

CHAPTER 1

About This Guide

BridgeLink is Innovar Healthcare's integration platform. It moves messages between the systems in your organization and lets you build, run and watch those connections, called channels, from a web browser.

The steps here guide you through downloading, installing and running a BridgeLink 26.9.0 server on a Mac, managed in WebAdmin.

A Mac is suitable for evaluation, development and single-user work. For a production BridgeLink server, use Windows or Linux.

If you would like a hand with any part of the install, Innovar Healthcare's professional services team can help with as much or as little of it as you want.

What You Install

Piece	What it is	Runs as	Default port
BridgeLink server	The integration platform itself: your channels and the service that runs them	A LaunchDaemon (a background service macOS starts and watches) under a dedicated, non-root account	8443 (HTTPS) and 8080 (HTTP)
WebAdmin	The browser-based tool you use to manage the server	The LaunchDaemon <code>com.innovar.bridgelink-webadmin</code> , as root	8444 (HTTPS)
Database	Where BridgeLink keeps channels, messages, events, users and settings	Your own PostgreSQL, MySQL, SQL Server or Oracle database	Set by your database

HTTPS is the encrypted form of a web connection.

Open port **8443** on the BridgeLink server to other machines only if WebAdmin runs on another machine or other programs call the BridgeLink API. Channels that listen for connections, such as MLLP or HTTP listeners, need their own ports opened. Port **8080** is BridgeLink's plain-HTTP port; nothing in this guide uses it, so keep it closed to other machines.

Decide These First

- **Which database BridgeLink uses.** Use your own external database, not the default Derby database. See [Choosing and Connecting the Database](#).
- **Which account runs BridgeLink in the background.** Use a dedicated service account, never root. See [The Service Account](#).

Requirements

- A Mac with Apple silicon. The BridgeLink macOS packages are built for Apple silicon only. Intel Macs are not supported.
- An administrator account on the Mac, to install software
- A Java runtime for Apple silicon. **Before You Install** in **Installing BridgeLink** says which version and builds to use.
- An external database with an empty database for BridgeLink
- For WebAdmin: Node.js 22, which the WebAdmin installer can install for you through Homebrew (a widely used tool for installing software on a Mac). Homebrew is not installed on a Mac by default; install it first from <https://brew.sh>.
- A current browser for WebAdmin: Chrome, Edge or Firefox 120 or later, or Safari 17 or later

CHAPTER 2

Getting the Software

BridgeLink downloads are on the BridgeLink website. Customers with an Innovar Healthcare customer portal account can also download them from the portal.

1. Open <https://www.bridgelink.net/download> in a browser.
2. Select **macOS**.
3. Find the **BridgeLink Server** card and confirm it shows version **26.9.0**.
4. Download the package you chose from the table below.

You do not need a separate WebAdmin download. Both packages put the WebAdmin installer, **BridgeLink-WebAdmin-install.tar.gz**, in the BridgeLink folder. You run it after BridgeLink is installed.

Installer or Archive

Package	What you get	Use it when
Installer (.dmg)	A guided setup that installs to /Applications/BridgeLink and asks for the ports. Apple silicon only.	Almost every install
Archive (.tgz)	The BridgeLink files only. You add Java and the background service yourself, with the steps below.	You want full control of where BridgeLink lives and how it runs

Neither package includes Java. Install Java first, as **Before You Install** in **Installing BridgeLink** describes.

CHAPTER 3

Installing BridgeLink

The commands in this guide use `/Applications/BridgeLink`, the installer's default folder. If you install somewhere else, use your folder in every command.

Before You Install

Neither package includes Java, so install it first. BridgeLink needs a Java runtime for Apple silicon: Java 21 (recommended) or Java 17. Use an OpenJDK build such as Eclipse Temurin (from Adoptium), Amazon Corretto or Azul Zulu. Java 17 is also supported when BridgeLink uses an external database; the default Derby database needs Java 21. Do not install a Java newer than 21: BridgeLink 26.9 does not use one.

1. Install a Java 21 runtime, using the vendor's Apple silicon (`aarch64`) `.pkg` installer. With Homebrew, `brew install --cask temurin@21` does the same. Do not use `brew install openjdk`: it installs the newest Java, which BridgeLink 26.9 does not use.
2. Open **Terminal** and list the Java runtimes installed on the Mac:

```
/usr/libexec/java_home -V
```

3. Look for a line that starts with `21.`, after its leading spaces. If no line starts with `21.`, Java 21 is not installed yet: install it as in step 1, then run the command again.

Installer (.dmg)

1. Double-click the downloaded `BridgeLink_macos_26_9_0.dmg` file. A window opens that shows **BridgeLink Installer**.
2. Double-click **BridgeLink Installer**.
3. macOS shows "'BridgeLink Installer' Not Opened". This happens on every Mac, because the installer is not yet notarized by Apple. Click **Done**.
4. Open **System Settings**, then click **Privacy & Security**.
5. Scroll to **Security** and click **Open Anyway** beside the BridgeLink Installer message.
6. Click **Open Anyway** again in the dialog that appears.
7. Enter your administrator password when macOS asks. The installer opens. You do not need to open the `.dmg` again.
8. Keep the install folder, `/Applications/BridgeLink`. If the folder shows anything else, change it to `/Applications/BridgeLink`.
9. On **BridgeLink Administrator Ports**, keep **Administrator Port** at `8443` and **Web Start Port** at `8080`.
10. On **Install Plugins to BridgeLink**, continue without adding any.
11. On the last screen, clear **Run BridgeLink**.

12. Click **Finish**.

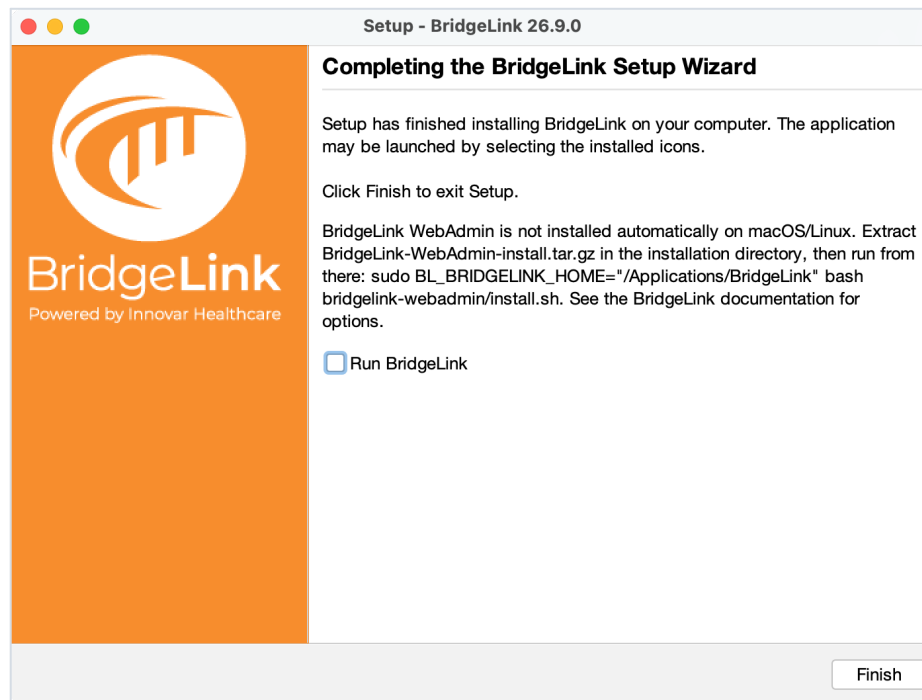


Figure 1: The BridgeLink installer on macOS at its last screen, showing the Run BridgeLink checkbox cleared and the Finish button.

Archive (.tgz)

1. In **Terminal**, extract the archive into **/Applications**. Safari's default setting, **Open safe files after downloading**, saves the file as **BridgeLink_macos_26_9_0.tar**, so the command uses that name. **tar** on macOS detects the compression by itself:

```
sudo tar -xf ~/Downloads/BridgeLink_macos_26_9_0.tar -C /Applications
```

If your browser kept the **.tgz** name, use **BridgeLink_macos_26_9_0.tgz** in the command instead.

2. Confirm the folder is **/Applications/BridgeLink**:

```
ls /Applications/BridgeLink
```

Whichever package you used, connect the database next. Do not start BridgeLink yet.

CHAPTER 4

Choosing and Connecting the Database

Every BridgeLink server keeps its channels, messages, events, users and settings in one database. You choose which database in

`/Applications/BridgeLink/conf/mirth.properties`, and you make that choice before BridgeLink starts for the first time.

Use an External Database, Not Derby

BridgeLink ships set to the default Derby database. Derby needs no setup, which makes it handy for a short test. Do not use it for a server you plan to keep:

- **Requires Java 21.** BridgeLink 26.9 will not start a server on the default Derby database with any Java older than 21. It stops at once and writes this line to `logs/mirth.log`:
`embedded Derby requires Java 21+ as of 26.9; upgrade Java or switch to an external database.` No setting turns this check off.
- **Runs inside BridgeLink.** Derby lives in the same process as the server, so no other tool can open the database while BridgeLink runs, and backing it up means copying its files.
- **Lacks features.** The default Derby database does not support the JSON value type in **Lookup Manager** or regex search in **Message Browser**. SQL Server does not support regex search either.
- **Sized for evaluation.** For production message volumes, use PostgreSQL, MySQL, Microsoft SQL Server or Oracle.

Prepare the Database

1. On your database server, create an empty database for BridgeLink, for example `bridgelinkdb`.
2. Create a login for BridgeLink that owns that database and can create tables in it.

BridgeLink creates all of its tables the first time it starts. You do not run any scripts.

Point BridgeLink at the Database

1. In a **Terminal** window, open the settings file in a terminal text editor as an administrator:

```
sudo nano /Applications/BridgeLink/conf/mirth.properties
```

If you installed with the `.dmg`, you can edit the file in TextEdit or another plain-text editor, such as BBEdit or Visual Studio Code, instead. In Finder, choose **Go > Go to Folder**, type `/Applications/BridgeLink/conf`, and open `mirth.properties`. The folder still belongs to the account that ran the installer, so you do not need `sudo` yet. After the archive install the file belongs to `root`, so use `sudo nano`: TextEdit would save a copy that

BridgeLink never reads.

- Find the line `database = derby` and change `derby` to the value for your database from the table below.
- Change the `database.url` line to your database's address, using the pattern in the table.
- Set `database.username` to the BridgeLink login.
- Set `database.password` to that login's password.
- Save the file. In nano, press **Control-O**, then **Return**.
- Close the editor. In nano, press **Control-X**.

Database	database value	database.url pattern
PostgreSQL	postgres	<code>jdbc:postgresql://dbhost:5432/bridgelinkdb</code>
MySQL	mysql	<code>jdbc:mysql://dbhost:3306/bridgelinkdb</code>
Microsoft SQL Server	sqlserver	<code>jdbc:sqlserver://dbhost:1433;databaseName=bridgelinkdb</code>
Oracle	oracle	<code>jdbc:oracle:thin:@//dbhost:1521/serviceName</code> for a service name (pluggable databases), or <code>jdbc:oracle:thin:@dbhost:1521:SID</code> for a SID

SQL Server and certificates. BridgeLink 26.9 connects to SQL Server with encryption on and checks the server's certificate. If your SQL Server's certificate is self-signed, comes from an internal certificate authority, or does not match the host name in the URL, BridgeLink cannot connect. Either add that certificate, or its certificate authority, to the Java trust store BridgeLink uses, or add `;trustServerCertificate=true` to the end of the URL. That keeps the connection encrypted but does not check which server answered.

Replace `dbhost`, the port and the database name (for Oracle, the service name or SID) with your own. A finished PostgreSQL example looks like this:

```
database = postgres
database.url = jdbc:postgresql://db01.example.org:5432/bridgelinkdb
database.username = bridgelink
database.password = your-password-here
```

Protect this file. `mirth.properties` holds the database password, so only administrators and the service account should be able to read it. The service account must also be able to write to it: on first start, BridgeLink writes new keystore passwords into the file. **The Service Account** sets this up, making the service account the file's owner and the only account that can read it, apart from administrators using `sudo`.

Leave the other lines as they are for now. The `https.port` and `http.port` lines hold your ports, and the `keystore` lines are managed by BridgeLink.

CHAPTER 5

The Service Account

To keep BridgeLink running after you sign out and start it when the Mac starts, run it as a LaunchDaemon, a background service that macOS manages. The daemon runs under one account, and that account decides what BridgeLink can reach.

Why Not Root

Channels can run scripts. If BridgeLink runs as **root**, any script in any channel has full control of the Mac. So BridgeLink refuses to start as **root** by default, and the log shows:

```
ERROR: BridgeLink is running as root/Administrator.
```

The setting behind this is `server.allowRoot = false` in `conf/mirth.properties`. Leave it set to **false**. A LaunchDaemon runs as **root** unless you name another account, so the file below names one.

Install as an Administrator, Run as the Service Account

You install BridgeLink with an administrator account: the installer asks for an administrator password, and the archive steps use **sudo**. After that, BridgeLink runs only as the service account this chapter creates, through the LaunchDaemon.

Do not start BridgeLink by typing `./blserver` in **Terminal** as yourself. Any file it creates, such as a log, then belongs to your account, and the service account may not be able to write to it later. For a quick test, run it as the service account with `sudo -u _bridgelinek`, as described in **Create the LaunchDaemon**. If you already ran it as yourself, run the commands in **Give It the BridgeLink Folder** again.

The service account needs:

- Ownership of the BridgeLink folder, because BridgeLink writes its logs, its **appdata** folder and `conf/mirth.properties` there
- Read and write access to any folders your channels read from or write to
- Network access to the database and to the systems your channels connect to

Create the Account

Run the commands in this chapter in a **Terminal** window.

Create a group for the service account first, so the account does not share the broad **staff** group:

```
sudo dseditgroup -o create -r "BridgeLink Service" -i 450 _bridgelinek
```

Then create a hidden service account in that group. It has no password and cannot sign in to the Mac, which is what you want for a background service:

```
sudo sysadminctl -addUser _bridgelink -roleAccount -UID 450 -GID 450
```

This creates the group `_bridgelink` with group ID `450` and the account `_bridgelink` with user ID `450`, with no home folder. `sysadminctl` requires the name of a role account (`-roleAccount`) to start with an underscore, and requires a user ID from `450` to `499`.

If another account or group already uses ID `450`, pick a different unused number from `450` to `499` and use it in both commands. These commands print the account and the group that have `450`, or nothing if it is free:

```
dscl . -list /Users UniqueID | awk '$2 == 450'
dscl . -list /Groups PrimaryGroupID | awk '$2 == 450'
```

Check the account:

```
dscl . -read /Users/_bridgelink UserShell NFSHomeDirectory IsHidden
```

The output shows `UserShell: /usr/bin/false`, which means the account cannot start a shell or sign in. It also shows `NFSHomeDirectory: /var/empty`, macOS's placeholder for an account with no home folder, and `dsAttrTypeNative:IsHidden: 1`, which keeps the account out of the sign-in screen.

Give It the BridgeLink Folder

```
sudo chown -R _bridgelink:_bridgelink /Applications/BridgeLink
sudo chmod -R go-w /Applications/BridgeLink
sudo chmod 600 /Applications/BridgeLink/conf/mirth.properties
```

The first command makes the service account the owner. The second removes write access for everyone else. The third lets only the service account read the file that holds the database password.

When you finish this chapter, the BridgeLink files look like this:

Path	Owner, group and permissions	Why
/Applications/BridgeLink and everything in it	_bridgelink:_bridgelink, and only the owner can write	BridgeLink writes its logs, its <code>appdata</code> folder and its settings here

Path	Owner, group and permissions	Why
/Applications/BridgeLink/conf/mirth.properties	_bridgelink:_bridgelink, 600	Only the service account can read it, and BridgeLink writes new keystore passwords into it on first start
/Library/LaunchDaemons/com.innovarhealthcare.bridgelink.plist	root:wheel, 644	macOS loads a LaunchDaemon file only when root owns it

WebAdmin's files are set up by its own installer. See [What the WebAdmin Installer Sets Up](#) in [Installing WebAdmin](#).

Create the LaunchDaemon

1. Create the file /Library/LaunchDaemons/com.innovarhealthcare.bridgelink.plist:

```
sudo nano /Library/LaunchDaemons/com.innovarhealthcare.bridgelink.plist
```

2. Paste this content:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
↳ "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.innovarhealthcare.bridgelink</string>
  <key>UserName</key>
  <string>_bridgelink</string>
  <key>WorkingDirectory</key>
  <string>/Applications/BridgeLink</string>
  <key>ProgramArguments</key>
  <array>
    <string>/Applications/BridgeLink/blserver</string>
  </array>
  <key>RunAtLoad</key>
  <true/>
  <key>KeepAlive</key>
  <dict>
    <key>SuccessfulExit</key>
    <false/>
  </dict>
  <key>Umask</key>
  <integer>23</integer>
  <key>StandardOutPath</key>
  <string>/Applications/BridgeLink/logs/launchd-stdout.log</string>
  <key>StandardErrorPath</key>
  <string>/Applications/BridgeLink/logs/launchd-stderr.log</string>
</dict>
</plist>
```

3. Save and close the editor.
4. Check the file for typing mistakes. The command prints OK when the file is valid:

```
plutil -lint /Library/LaunchDaemons/com.innovarhealthcare.bridgelink.plist
```

5. Set the file's owner and permissions, which macOS requires for a LaunchDaemon:

```
sudo chown root:wheel \
  /Library/LaunchDaemons/com.innovarhealthcare.bridgelink.plist
sudo chmod 644 /Library/LaunchDaemons/com.innovarhealthcare.bridgelink.plist
```

The file runs **blserver**, BridgeLink's own launcher. It finds the Java 21 you installed and reads BridgeLink's Java options from **/Applications/BridgeLink/blserver.voptions**, so the file needs no Java path. Remove any Java newer than 21: with a newer Java installed, BridgeLink can fall back to an older Java such as 17 even when 21 is installed.

The **KeepAlive** setting with **SuccessfulExit** set to **false** makes macOS start BridgeLink again when it stops with an error. It does not restart BridgeLink after a clean stop, such as **launchctl bootout**. Because a failing BridgeLink keeps coming back, a running service does not prove that BridgeLink started. **Starting BridgeLink for the First Time** shows the check to use.

The **Umask** value **23** is the decimal form of **027**. It keeps the files BridgeLink creates, such as its logs, unreadable by accounts other than the owner and its group.

Do not load it yet. **Starting BridgeLink for the First Time** does that.

To run BridgeLink by hand as the service account, for a quick test, use **sudo -u _bridgelinek**, not **su**. The account cannot sign in on purpose, so **su - _bridgelinek** always fails.

Check Which Account BridgeLink Runs As

After BridgeLink starts, confirm it is running as the service account:

```
ps -o user,pid,command -ax | grep "[m]irth-server-launcher"
```

The first column is the account. It should read **_bridgelinek**.

Do not use **lsuf** for this check. Without **sudo**, **lsuf** lists only your own processes, so it shows nothing for a BridgeLink server running as another account, even while that server is working normally.

Changing the Account After Install

If BridgeLink runs as one account and you switch it to another, the files BridgeLink writes still belong to the old account. The new account cannot write its logs, its keystore or its settings, and BridgeLink fails with **Permission denied** errors.

First, create the new account if it does not exist yet, the same way as in **Create the Account** above, with its own name, its own group and an unused ID. Then:

1. Stop BridgeLink:

```
sudo launchctl bootout system/com.innovarhealthcare.bridgelinek
```

2. Open the daemon file and change the **UserName** value to the new account:

```
sudo nano /Library/LaunchDaemons/com.innovarhealthcare.bridgelinek.plist
```

3. Give the new account the BridgeLink folder:

```
sudo chown -R newaccount:newaccount /Applications/BridgeLink
sudo chmod -R go-w /Applications/BridgeLink
sudo chmod 600 /Applications/BridgeLink/conf/mirth.properties
```

4. Give the new account the same access to any folders your channels use.
5. Start BridgeLink again:

```
sudo launchctl bootstrap system \
/Library/LaunchDaemons/com.innovarhealthcare.bridgelink.plist
```

6. Check the account with the `ps` command above.

CHAPTER 6

Starting BridgeLink for the First Time

1. Load the LaunchDaemon, which starts BridgeLink now and at every restart:

```
sudo launchctl bootstrap system \
/Library/LaunchDaemons/com.innovarhealthcare.bridgelink.plist
```

2. Wait about a minute.
3. Check that the BridgeLink web server started. The log files are not readable by every account, so use **sudo**:

```
sudo grep 'Web server running at' \
/Applications/BridgeLink/logs/mirth.log | tail -1
```

4. Confirm the command printed a line, and that the time on it is after the start you just made. That line is the sign that BridgeLink started. If nothing prints, or the newest line is from an earlier start, BridgeLink did not start. See [Troubleshooting](#).
5. Find the newest line that contains **Running**, to check the database name and Java version:

```
sudo grep 'Running ' /Applications/BridgeLink/logs/mirth.log | tail -1
```

6. Confirm the line names your database, not **derby**. It also names the Java version, which should be 21.
7. Confirm BridgeLink runs as **_bridgelink**, using the **ps** check in [The Service Account](#).

Do not rely on `launchctl print` showing `state = running`. macOS restarts BridgeLink when it fails, so the service can show as running while BridgeLink keeps failing. The **Running** line alone is not enough either: it can appear even when the web server did not start. After you fix a failure and BridgeLink starts, `launchctl print` can still show `last exit code = 1`. That leftover value means nothing once the **Web server running at** line appears.

To restart BridgeLink later, run `sudo launchctl kickstart -k system/com.innovarhealthcare.bridgelink`.

On this first start, BridgeLink:

- Creates its tables in your database
- Creates a keystore holding a new self-signed certificate for port **8443**, and writes new keystore passwords into `conf/mirth.properties`
- Creates the `appdata/server.id` file that holds this server's unique ID

If BridgeLink stops right away, see [Troubleshooting](#).

Memory for the BridgeLink Server

BridgeLink starts with a 256 MB Java heap, which is enough for a personal or test install but small for production. The `-Xmx256m` line in `/Applications/BridgeLink/blserver.vmoptions` sets it. For heavier use, size it to the message volume and leave memory for macOS and the database.

To change it:

1. Open the file as an administrator:

```
sudo nano /Applications/BridgeLink/blserver.vmoptions
```

2. Change the `-Xmx256m` line, for example to `-Xmx1g`.
3. Save the file and close the editor.
4. Restart BridgeLink:

```
sudo launchctl kickstart -k system/com.innovarhealthcare.bridgelink
```

CHAPTER 7

Installing WebAdmin

WebAdmin is the browser-based tool you use to manage BridgeLink. It runs as its own small service, next to BridgeLink. WebAdmin is a small web application run by Node.js: it serves the pages you see in the browser and passes each request on to BridgeLink. It contains no Java.

On macOS, WebAdmin is a separate step with its own installer. You run it after BridgeLink is installed, from the `BridgeLink-WebAdmin-install.tar.gz` archive in the BridgeLink folder. The installer's last screen points to it too.

Run the WebAdmin Installer

WebAdmin needs Node.js 22, and the installer checks. If it finds no Node.js, or a different version such as Homebrew's newer `node`, it offers to install Node.js 22 through Homebrew. Homebrew must already be installed (see <https://brew.sh>). On a Mac that already has a newer `node`, Node.js 22 goes in next to it and does not replace it. Homebrew then prints hints such as "To link this version, run: `brew link node@22`" and a `PATH` line for `~/ .zshrc`. Ignore them: following them replaces the Mac's own `node`. On a Mac with no Node.js, option 1 also makes Node.js 22 the Mac's `node` command. In both cases WebAdmin runs `/opt/homebrew/opt/node@22/bin/node`.

1. In **Terminal**, go to the BridgeLink folder:

```
cd /Applications/BridgeLink
```

2. Extract the WebAdmin archive there. It creates a `bridgelink-webadmin` folder:

```
sudo tar -xzf BridgeLink-WebAdmin-install.tar.gz
```

3. Run the installer from the BridgeLink folder, telling it where BridgeLink is so **Same Host (Recommended)** can find BridgeLink's port:

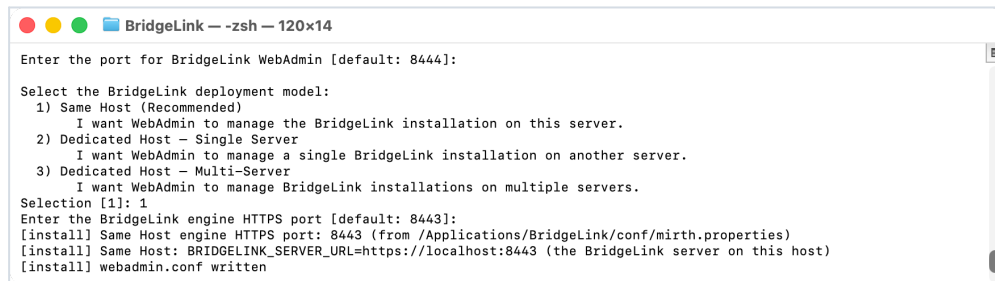
```
sudo BL_BRIDGELINK_HOME=/Applications/BridgeLink \  
bash bridgelink-webadmin/install.sh
```

To use a Node.js 22 you already have through Homebrew (`node@22`), run this command instead:

```
sudo BL_BRIDGELINK_HOME=/Applications/BridgeLink \  
NODEJS_HOME="$(brew --prefix node@22)" \  
bash bridgelink-webadmin/install.sh
```

If your Node.js 22 is not Homebrew's `node@22`, set `NODEJS_HOME` to the folder that contains your Node.js 22's `bin/node`, for example `~/.nvm/versions/node/v22.x.y`, instead of `"$(brew --prefix node@22)"`.

4. Read the license terms it points to, `LICENSE` and `SUPPLEMENTAL-TERMS.txt`.
5. At `Do you accept the license terms? [y/N]:`, type `y`.
6. If the installer shows `Choose an option:` for Node.js, press `Return` to choose 1) `Install Node.js 22 now`.
7. Confirm any prompt macOS shows.
8. At `Enter the port for BridgeLink WebAdmin [default: 8444]:`, press `Return`.
9. At `Select the BridgeLink deployment model:`, type `1` for `Same Host (Recommended)`.
10. At `Enter the BridgeLink engine HTTPS port [default: 8443]:`, press `Return`, unless you changed BridgeLink's HTTPS port. If you did, type that port.



```

BridgeLink -- zsh -- 120x14
Enter the port for BridgeLink WebAdmin [default: 8444]:

Select the BridgeLink deployment model:
  1) Same Host (Recommended)
      I want WebAdmin to manage the BridgeLink installation on this server.
  2) Dedicated Host - Single Server
      I want WebAdmin to manage a single BridgeLink installation on another server.
  3) Dedicated Host - Multi-Server
      I want WebAdmin to manage BridgeLink installations on multiple servers.
Selection [1]: 1
Enter the BridgeLink engine HTTPS port [default: 8443]:
[install] Same Host engine HTTPS port: 8443 (from /Applications/BridgeLink/conf/mirth.properties)
[install] Same Host: BRIDGELINK_SERVER_URL=https://localhost:8443 (the BridgeLink server on this host)
[install] webadmin.conf written
  
```

Figure 2: A macOS Terminal window running the WebAdmin install.sh, showing the WebAdmin port prompt answered with the default 8444, the deployment model menu with 1 entered, and the engine HTTPS port prompt with the port read from `mirth.properties`.

While it creates the self-signed certificate, the installer prints a long run of `.+...+++*` characters. This is normal. Wait for it to finish.

The installer:

- Copies WebAdmin to `/opt/bridgelink-webadmin` and writes `/opt/bridgelink-webadmin/webadmin.conf`
- Creates a self-signed certificate for WebAdmin at `/opt/bridgelink-webadmin/cert.pem` and `key.pem`
- Installs and loads the LaunchDaemon `com.innovar.bridgelink-webadmin`, which starts at boot and restarts if it stops
- Installs the `blwebadmin` command for day-to-day control: `blwebadmin status`, `sudo blwebadmin restart`, `blwebadmin logs`

`blwebadmin status` prints about 60 lines of `launchctl print` output. The line to look for is `state = running`.

Same Host finds BridgeLink's port. Same Host (Recommended) reads BridgeLink's HTTPS port from `/Applications/BridgeLink/conf/mirth.properties` and uses 8443 only if it cannot find it, with a warning. If you see that warning and BridgeLink uses another port, open the file with `sudo nano /opt/bridgelink-webadmin/webadmin.conf`, correct the port in `BRIDGELINK_SERVER_URL`, save it, and run `sudo blwebadmin restart`.

What the WebAdmin Installer Sets Up

The WebAdmin service runs as `root`, and everything the installer creates belongs to `root`, so stopping or restarting WebAdmin takes `sudo`. macOS restarts WebAdmin whenever it stops, so control it only with `sudo blwebadmin restart`, `sudo blwebadmin stop` and `sudo blwebadmin start`. Do not run `start-service.sh` yourself: a second copy cannot use port 8444 and fails with `EADDRINUSE`.

Errors go to `/var/log/bridgelink-webadmin.err`, not to the `.log` file that `blwebadmin logs` follows:

```
sudo tail -40 /var/log/bridgelink-webadmin.err
```

The **BridgeLink WebAdmin Setup Guide** lists every file the installer creates, with its owner and permissions, and the log lines that are normal.

Sign In for the First Time

1. Open `https://localhost:8444` in a browser.
2. If the browser shows a certificate warning, click **Advanced**, then **Proceed to ... (unsafe)** in Chrome or **Continue to ... (unsafe)** in Edge. In Firefox, click **Advanced**, then **Accept the Risk and Continue**. In Safari, click **Show Details**, click **visit this website**, then click **Visit Website** in the confirmation dialog. Read the note after this step.

When a browser stops letting you click through. WebAdmin tells browsers to accept only a valid certificate for its server name and every name under it, on every port, for two years. This is called HSTS. A browser starts enforcing it the first time it opens WebAdmin with a certificate it trusts, for example after you install your own certificate (see [Replacing the Certificates](#)). Clicking through a warning does not start it. `localhost` and IP addresses are never affected. Once a browser enforces HSTS for a name, it also refuses BridgeLink's default certificate on port 8443 at that name.

3. In **Username**, type `admin`.
4. In **Password**, type `admin`.
5. Click **Login**.
6. On **Welcome to BridgeLink**, leave **Username** as `admin`.

7. In **New Password**, type a new password of at least 8 characters, with an uppercase letter, a lowercase letter, a number, and a special character such as **!, @, #, \$, %** or *****. A hyphen, period or space does not count as a special character, and the password cannot be **admin**. WebAdmin lists only the rules that **admin** breaks, so check your password against all of these.
8. Type it again in **Confirm Password**.
9. The profile fields on **Welcome to BridgeLink** are optional; fill them in if you want to.
10. Click **Save & Continue**.

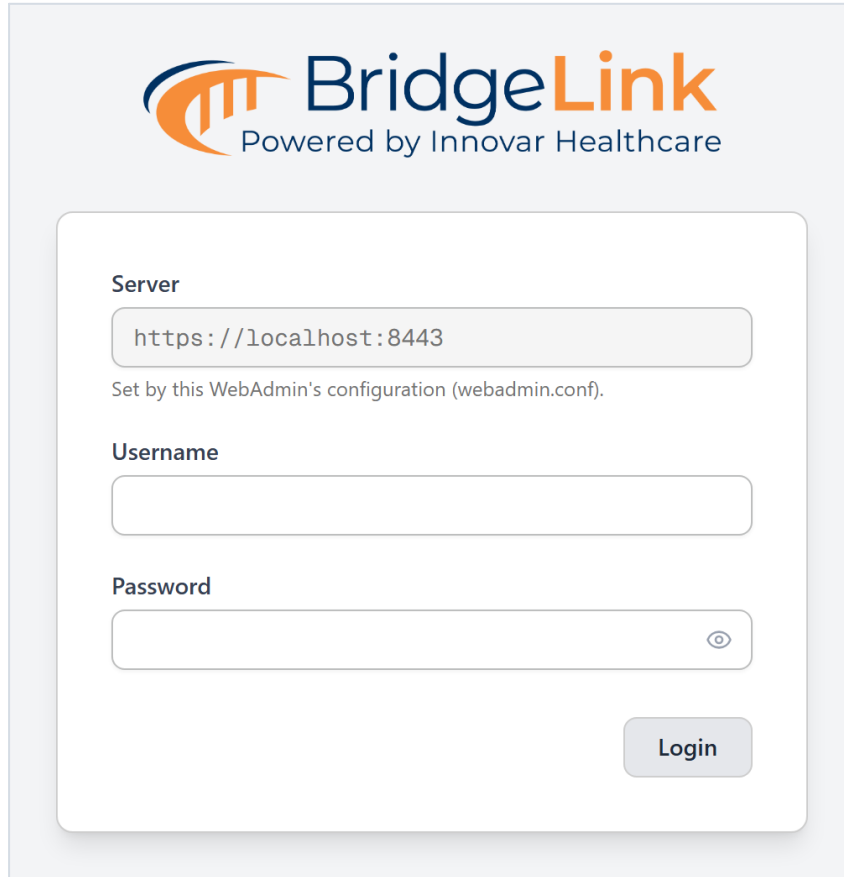


Figure 3: The WebAdmin sign-in screen in a browser, showing the Server field filled in and read-only for a Same Host install, the Username and Password fields, and the Login button.

With **Same Host (Recommended)**, the **Server** field is filled in and cannot be changed: WebAdmin talks only to the BridgeLink server on this Mac, at **localhost**. To point it somewhere else, change **BRIDGELINK_SERVER_URL** in **/opt/bridgelink-webadmin/webadmin.conf** and run **sudo blwebadmin restart**.

More WebAdmin Setups

For WebAdmin on another machine, one WebAdmin for several servers, your own certificate, and every setting, see the **BridgeLink WebAdmin Setup Guide** on the BridgeLink download page, <https://www.bridgelink.net/download>.

CHAPTER 8

Finding the Server ID

Every BridgeLink server has a unique server ID, created the first time it starts. Plugin licenses are issued against this ID, so Innovar Healthcare asks for it when you license a plugin.

In WebAdmin

1. Sign in to WebAdmin.
2. In the left sidebar, scroll to the bottom and, under **Help**, click **About BridgeLink**.
If **Update Available** opens instead, click **Got it**, then click **About BridgeLink** again.
3. Click the copy icon beside **Server ID**.

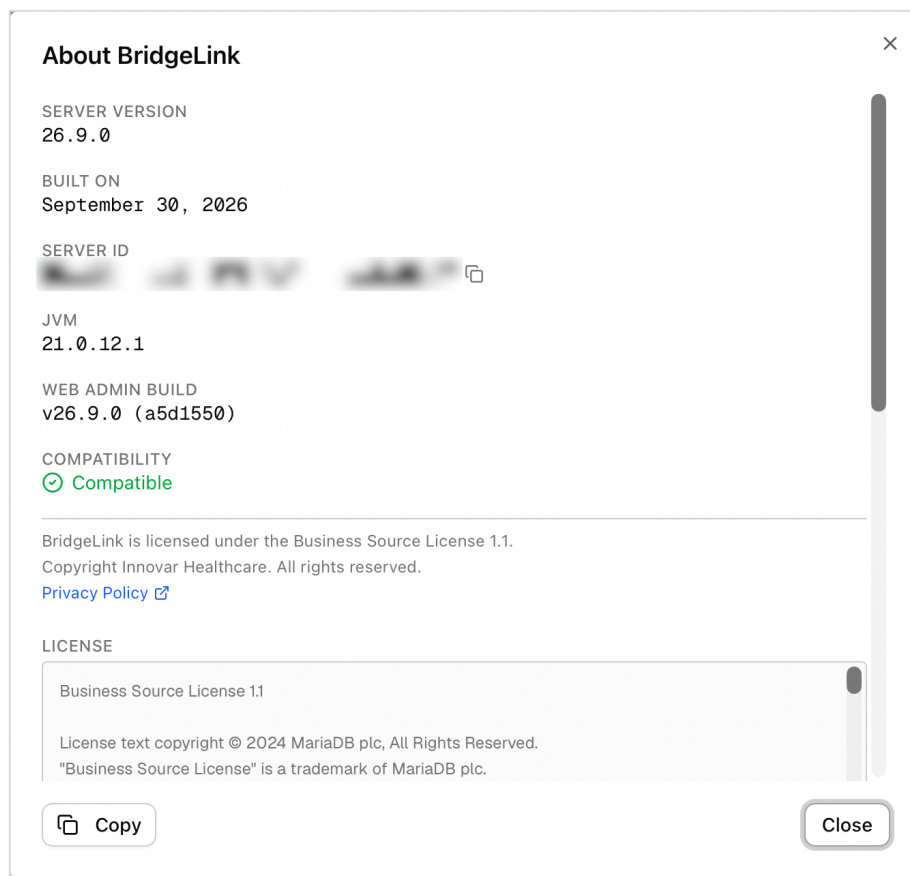


Figure 4: The About BridgeLink dialog in WebAdmin, showing the Server Version, Built On, Server ID (blurred here, with its copy icon), JVM, Web Admin Build and Compatibility rows.

On Disk

The same ID is in `/Applications/BridgeLink/appdata/server.id`, on a line that reads `server.id =` followed by the ID:

```
sudo cat /Applications/BridgeLink/appdata/server.id
```

Back up `/Applications/BridgeLink/appdata/server.id`, `/Applications/BridgeLink/appdata/keystore.jks` and `/Applications/BridgeLink/conf/mirth.properties` together. The keystore holds the key BridgeLink uses to encrypt data, and `mirth.properties` holds the keystore's passwords. If you rebuild the Mac and put `server.id` back, owned by the service account, before BridgeLink first starts, the server keeps its ID and its plugin licenses. Without the file, BridgeLink creates a new ID.

CHAPTER 9

Replacing the Certificates

A new install uses two self-signed certificates (certificates the server creates for itself, which browsers do not trust), which is why browsers warn the first time. If other users will connect to this Mac, replace them with certificates from your organization's certificate authority (CA).

Certificate	Port	Who sees it	Where it lives
WebAdmin	8444	Every browser that opens WebAdmin	/opt/bridgelink-webadmin/cert.pem and key.pem
BridgeLink server	8443	WebAdmin and any program that calls the BridgeLink API	The mirthconnect entry in /Applications/BridgeLink/appdata/keystore.jks

Replace the WebAdmin Certificate

Before you replace WebAdmin's certificate, read **When a browser stops letting you click through** under **Sign In for the First Time** in **Installing WebAdmin**.

You need the certificate and its private key as PEM files (text files that begin with -----BEGIN). Put any intermediate certificates in the certificate file after the server certificate. The private key file must not have a password.

1. Stop WebAdmin:

```
sudo blwebadmin stop
```

2. Keep copies of the current files:

```
sudo cp /opt/bridgelink-webadmin/cert.pem /opt/bridgelink-webadmin/cert.pem.bak
sudo cp /opt/bridgelink-webadmin/key.pem /opt/bridgelink-webadmin/key.pem.bak
```

3. Copy your certificate over cert.pem:

```
sudo cp your-certificate.pem /opt/bridgelink-webadmin/cert.pem
```

4. Copy your private key over key.pem:

```
sudo cp your-private-key.pem /opt/bridgelink-webadmin/key.pem
```

5. Start WebAdmin:

```
sudo blwebadmin start
```

- Open WebAdmin in a browser and confirm there is no certificate warning.

To Undo the Swap

If browsers cannot open WebAdmin after the swap, put the old certificate back:

- Stop WebAdmin:

```
sudo blwebadmin stop
```

- Copy the two .bak files back:

```
sudo cp /opt/bridgelink-webadmin/cert.pem.bak /opt/bridgelink-webadmin/cert.pem
sudo cp /opt/bridgelink-webadmin/key.pem.bak /opt/bridgelink-webadmin/key.pem
```

- Start WebAdmin:

```
sudo blwebadmin start
```

Browsers that opened WebAdmin by its server name with the new certificate refuse the old self-signed one. Fix the new certificate instead, or clear the browser's HSTS entry for the server name: open `chrome://net-internals/#hsts` in Chrome or `edge://net-internals/#hsts` in Edge, and use **Delete domain security policies**. In other browsers, clear the browsing data for that server name.

Replace the BridgeLink Server Certificate

BridgeLink keeps its certificate in a keystore file under the name `mirthconnect`. The same keystore also holds the key BridgeLink uses to encrypt data, so replace only the `mirthconnect` entry. Never swap in a different keystore file.

You need the certificate and private key as a PKCS12 file (a single file, ending in `.p12` or `.pfx`, that bundles a certificate with its private key) and its password. The commands use `keytool`, which comes with Java.

Steps 5 and 6 use the values in this table:

Placeholder	Where its value comes from
<your .p12 file password>	The password of your .p12 or .pfx file
<the entry name from step 4>	The entry name that step 4 listed
<your keystore.storepass value>	The keystore.storepass value that step 3 showed
<your keystore.keypass value>	The keystore.keypass value that step 3 showed

- Stop BridgeLink:

```
sudo launchctl bootout system/com.innovarhealthcare.bridgelink
```

2. Back up the keystore:

```
sudo cp /Applications/BridgeLink/appdata/keystore.jks ~/keystore.jks.bak
sudo chmod 600 ~/keystore.jks.bak
```

3. Note the values of `keystore.storepass` and `keystore.keypass`. You use them in steps 5 and 6:

```
sudo grep '^keystore\.' /Applications/BridgeLink/conf/mirth.properties
```

4. List the entry name inside your PKCS12 file:

```
keytool -list -keystore server.p12 -storetype PKCS12
```

5. Remove the old certificate entry. Replace `<your keystore.storepass value>` with the `keystore.storepass` value that step 3 showed. Do this step before step 6:

```
sudo keytool -delete -alias mirthconnect \
  -keystore /Applications/BridgeLink/appdata/keystore.jks \
  -storetype JCEKS -storepass <your keystore.storepass value>
```

6. Import yours as `mirthconnect`. Replace the placeholders with your own values:

```
sudo keytool -importkeystore \
  -srckeystore server.p12 -srcstoretype PKCS12 \
  -srcstorepass <your .p12 file password> \
  -srcalias <the entry name from step 4> \
  -destkeystore /Applications/BridgeLink/appdata/keystore.jks \
  -deststoretype JCEKS \
  -deststorepass <your keystore.storepass value> \
  -destalias mirthconnect \
  -destkeypass <your keystore.keypass value>
```

Steps 5 and 6 print a warning that recommends migrating the keystore to PKCS12. Ignore it: BridgeLink reads the keystore as JCEKS, so do not convert it.

7. Give the keystore back to the service account:

```
sudo chown _bridgelink:_bridgelink /Applications/BridgeLink/appdata/keystore.jks
```

8. Start BridgeLink:

```
sudo launchctl bootstrap system \
  /Library/LaunchDaemons/com.innovarhealthcare.bridgelink.plist
```

9. Open <https://localhost:8443> in a browser and confirm the browser shows your certificate.

If BridgeLink does not start, or port 8443 does not answer: stop BridgeLink, run `sudo cp ~/keystore.jks.bak /Applications/BridgeLink/appdata/keystore.jks`, repeat the `chown` from step 7, and start BridgeLink.

WebAdmin keeps working through the change, with no change to its own settings.

CHAPTER 10

The Data Pruner

BridgeLink keeps the messages each channel processes until the Data Pruner removes them. The Data Pruner is off on a new server, so turn it on before you send real traffic through BridgeLink; otherwise the database grows with every message.

It has two parts, and both have to be set:

- **The schedule, for the whole server:** in WebAdmin, in the left sidebar under **Administration**, click **Settings**, then the **Data Pruner** tab
- **How long each channel keeps messages:** on each channel's **Summary** tab, under **Message Pruning**. A channel whose pruning fields are blank is never pruned, even when the schedule is on.

Choose ages that match your organization's record-keeping rules.

Turn On the Schedule

1. In WebAdmin, in the left sidebar under **Administration**, click **Settings**.
2. Click the **Data Pruner** tab.
3. Under **Schedule**, set **Enable** to **Yes**.
4. Choose a **Schedule Type** and set when it runs: **Interval** runs every set number of hours or minutes, **Time** runs once a day, and **Cron** takes cron expressions. Once a day at a quiet hour suits most servers.
5. Click **Save**.

Settings

Server Administrator Tags Configuration Map Resources **Data Pruner** Message Trends Version History

Status

Current State: **Not running**

Current Process: -

Last Process: -

Next Process: Scheduled Monday, Sep 28, 1:00:00 AM

Schedule

Enable: ☒ Yes ☐ No

Schedule Type: **Time**

Time (HH : MM): **1** : **0**

Prune Settings

Block Size: **1000** (50-10000, recommended: 1000)

Prune Events: ☒ Yes ☐ No

Prune Event Age: **90** days

Archive Settings

Enable Archiving: ☐ Yes ☒ No

Figure 5: The Data Pruner tab of the Settings page, showing the Status section, the Schedule section with Enable set to Yes and Schedule Type set to Time, the Prune Settings section and the Archive Settings section.

Set How Long Each Channel Keeps Messages

1. In the left sidebar under **Build**, click **Channels**.
2. Double-click the channel to open it in the channel editor.
3. On the **Summary** tab, under **Message Pruning**, enter the number of days in **Prune metadata after** and **Prune content after**. Content cannot be kept longer than metadata.
4. Click **Save**.
5. Repeat for each channel.

Message Pruning

Prune metadata after **30** days

Prune content after **7** days

☒ Archive messages before pruning

☐ Prune errored messages

(incomplete, errored, and queued messages will not be pruned)

Figure 6: The Message Pruning section of a channel's Summary tab, showing Prune metadata after set to 30 days, Prune content after set to 7 days, Archive messages before pruning checked and Prune errored messages cleared.

Incomplete and queued messages are never pruned. Errored messages are kept too, unless you check **Prune errored messages**.

CHAPTER 11

Troubleshooting

Symptom	Likely cause	What to do
macOS says "'BridgeLink Installer' Not Opened" when you open the installer in the disk image	Gatekeeper, the Mac's check on downloaded apps, could not verify the installer app, BridgeLink Installer	Click Done . Open System Settings , click Privacy & Security , scroll to Security , and click Open Anyway . Click Open Anyway again in the dialog, and enter your password. The installer then opens.
BridgeLink stops at once, and logs/mirth.log shows ERROR: BridgeLink is running as root/Administrator.	The LaunchDaemon has no UserName, so it runs as root	Add the UserName key. See Create the LaunchDaemon .
BridgeLink stops at once, and the log shows embedded Derby requires Java 21+ as of 26.9; upgrade Java or switch to an external database	The server is still on the default Derby database, running on Java older than 21	Connect an external database, or install Java 21. To check that Java 21 is installed, run <code>/usr/libexec/java_home -V</code> and look for a line that starts with 21.. If no line does, Java 21 is not installed. Remove any Java newer than 21: with a newer Java installed, BridgeLink can fall back to an older Java such as 17 even when 21 is installed.
BridgeLink does not start. <code>launchctl print</code> shows it running, the log shows Could not initialize security settings and appdata/keystore.jks (Permission denied), and there is no Web server running at line	The service account does not own the BridgeLink folder, often after an account change or after BridgeLink was started in Terminal as your own account	Run the three commands in Give It the BridgeLink Folder : <code>sudo chown -R _bridgelink:_bridgelink /Applications/BridgeLink</code> , <code>sudo chmod -R go-w /Applications/BridgeLink</code> and <code>sudo chmod 600 /Applications/BridgeLink/conf/mirth.properties</code> . Then run <code>sudo launchctl kickstart -k system/com.innovarhealthcare.bridgelink</code> . BridgeLink does not recover by itself.
BridgeLink does not write anything to logs/mirth.log and keeps starting over, for example with No suitable Java Virtual Machine could be found	The failure happened before BridgeLink's own logging started, so the message went to the LaunchDaemon's own logs. The launcher exits with code 83, and because KeepAlive restarts it, BridgeLink keeps starting over rather than stopping.	Read the end of both LaunchDaemon logs: <code>sudo tail -50 /Applications/BridgeLink/logs/launchd-stderr.log</code> and <code>sudo tail -50 /Applications/BridgeLink/logs/launchd-stdout.log</code> . The Java message is in <code>launchd-stdout.log</code> . Failures before BridgeLink's own logging, such as a missing Java, appear only in these logs. To confirm, run <code>sudo launchctl print system/com.innovarhealthcare.bridgelink</code> and look for last exit code = 83.
<code>launchctl print</code> shows state = running, but BridgeLink does not answer or keeps starting over	BridgeLink fails, and macOS restarts it each time	Read the newest lines with <code>sudo tail -50 /Applications/BridgeLink/logs/mirth.log</code> and fix what they name. BridgeLink has started only when a new Web server running at line appears. A leftover last exit code = 1 in <code>launchctl print</code> means nothing once BridgeLink has started.

Symptom	Likely cause	What to do
keytool prints Keystore was tampered with, or password was incorrect	The keystore password in the command is wrong, or a placeholder was left in the command	Use the keystore.storepass value from conf/mirth.properties, and replace every placeholder in the command with a real value. See Replace the BridgeLink Server Certificate .
WebAdmin is signed in, but the Dashboard shows Unknown server error	WebAdmin cannot reach BridgeLink	Check for a recent Web server running at line, as in Starting BridgeLink for the First Time . If it is missing, BridgeLink did not start.
The log shows the database connection failing at startup	The database.url, login or password is wrong, or a firewall blocks the database port. SQL Server only: its certificate is self-signed, comes from an internal certificate authority, or does not match the host name in the URL, which BridgeLink 26.9 does not accept by default.	Check the four database lines in conf/mirth.properties. BridgeLink retries twice, 10 seconds apart, before it gives up. See SQL Server and certificates in Choosing and Connecting the Database .
WebAdmin sign-in shows "Could not reach the BridgeLink server. Verify the server address is correct and the server is running."	BridgeLink is not running, or WebAdmin points at the wrong BridgeLink port	Start BridgeLink. Compare the port in BRIDGELINK_SERVER_URL in /opt/bridgelink-webadmin/webadmin.conf with BridgeLink's HTTPS port. If they differ, correct it and run sudo blwebadmin restart.
WebAdmin sign-in fails while BridgeLink is running, and /var/log/bridgelink-webadmin.err shows Upstream fetch failed for users/_login: TypeError: fetch failed	WebAdmin is running on a Node.js version other than 22	To see which Node.js WebAdmin uses, run sudo grep '^exec' /opt/bridgelink-webadmin/start-service.sh, then run that path with --version. If it is not v22, run the WebAdmin installer again (sudo BL_BRIDGELINK_HOME=/Applications/BridgeLink bash bridgelink-webadmin/install.sh from /Applications/BridgeLink), let it install Node.js 22, and answer the port and deployment model prompts as before.
EADDRINUSE: address already in use :::8444, and a WebAdmin process you stop comes straight back	WebAdmin is already running as its LaunchDaemon, and a second copy was started by hand	Do not start another copy. Use sudo blwebadmin restart. See What the WebAdmin Installer Sets Up .
The browser warns about the certificate	WebAdmin still uses its self-signed certificate	Click through the warning, or replace the certificate. See Replace the WebAdmin Certificate . If the browser offers no way to continue, it is enforcing HSTS for that name. See When a browser stops letting you click through under Sign In for the First Time , and To Undo the Swap in Replacing the Certificates to clear it.

For more WebAdmin problems, see [Troubleshooting](#) in the **BridgeLink WebAdmin Setup Guide**.

Support

Customers with an active Innovar Healthcare support plan can contact support at support@innovarhealthcare.com. Include the BridgeLink version, the package you installed, your macOS version, what you were doing, and the most recent part of [logs/mirth.log](#). [mirth.log](#) can contain message content, which may include patient data. Remove patient data before you send a log, or ask support for a secure way to send it.