BFQuarantine

USER MANUAL

BFQuarantine

Version 1.33.0

Put the robot you were sold next to the backtest it was sold with, on the account where it runs — MetaTrader 5, JForex or a web3 wallet — and check every week whether it still looks the same. Its code is never run: it reads what the account really did.

The same Help that lives inside the program, chapter by chapter. It is generated from the program's own source, so it can't say anything the program doesn't do.

Inside the program, this same Help lives under Help, with a search box (Ctrl+K), and Help → Take the tour walks through every screen in the order of the work.

BFQuarantine 1.33.0 · 26 October 2026

Contents

GETTING STARTED

What is BFQuarantine 3

Home 3

CONNECTING THE PLATFORM

Connecting MetaTrader 5 4

Connecting JForex 6

QUARANTINE AND LIVE

Quarantine 8

Live accounts and wallets 8

AT THE END

Frequently asked questions 10

Metrics glossary 10

Shortcuts and tips 10

GETTING STARTED

What is BFQuarantine

BFQuarantine holds a trading robot you were sold against the backtest it was sold with. Load the robot's file and, if you have it, its backtest; point it at the account where the robot runs; and a record builds up of the distance between what it promised and what the account really does. It only reads: the robot's code is never run, and nothing here can place an order. What it records is kept on this machine between sessions.

The menu on the left:

1. **Home** — the curve you are watching: a quarantine, or a live account.
2. **Quarantine** — the robot you were sold, held against its backtest on the account where it runs.
3. **Live** — the account's real trades on the chart, and the place to connect your MetaTrader 5, MetaTrader 4 or JForex.

Home

Landing screen. It draws a curve and lets you pick which: a Hyperliquid wallet's —what it has made trading, without the deposits and withdrawals—, if you added one in Settings, or each quarantine's —what the robot has made since it went in, one closed trade at a time, in its account's currency—. Under the title it says which of the two it is. Below, shortcuts to Quarantine and Live.

CONNECTING THE PLATFORM

Connecting MetaTrader 5

BFQuarantine reads the trades your robot made on a running MetaTrader 5 terminal, through a small add-in. It reads the terminal and **never places an order**.

MetaTrader 4 connects the same way, through a bridge of its own: the last section of this chapter, *Connecting MetaTrader 4*, tells how.

How it connects

Through a small add-in called the **connector**: an MQL5 *service* that runs inside MetaTrader with no chart. It talks to this app by writing files in MetaTrader's own common folder.

That is a deliberate choice. A network connection — MetaTrader's MCP server, a socket, or `WebRequest` — needs a setting inside the terminal that has to be switched on and remembered, and MetaTrader is known to forget it between restarts. Files need no port, no allow-list, no firewall rule and no dialog. And because a *service* is restarted by the terminal every time it opens, you install it once and never think about it again.

Installing it, once

1. Open MetaTrader 5 and log in to your account.
2. Here, in **Settings** → **Trading platforms**, press **Install the connector** in the MetaTrader 5 panel. It is also in **Live**, while no MetaTrader is connected: **Connect the MetaTrader 5 where your robot runs** opens it.
3. If MetaTrader was open, close it and open it again; if it was closed, just open it. The connector starts with it.

The button does all of it: it finds MetaTrader's folder —inside the Wine prefix too—, puts the connector in its services folder and sets it to start with the terminal, the same entry MetaTrader writes when you add a service by hand. From then on the terminal starts it every time it opens, and the panel here turns green on its own. If you would rather not close MetaTrader right now, you can start it from the Navigator instead (`Ctrl+N`): **Services** → **BotFactoryFX** → right-click → **Add service**, then **Start** — once: adding it twice makes two copies, and the panel says so.

The bridge folder

The connector is a small program of its own — **BotFactoryFX Bridge** — and it travels inside this app as a folder: the compiled service (`BotFactoryFX.ex5`), its source (`BotFactoryFX.mq5`, so anyone can read what it does), the icon MetaTrader shows in the Navigator, a `LEEME.txt` and the manual as a PDF. Under the install cards there are three buttons: **Save the bridge folder (connector, manual, icon)** copies the whole folder to `~/.botfactoryfx/conectores/metatrader5` and shows the path with a copy button; **Open the manual** saves it and opens the PDF; **Open the folder** saves it and opens that folder in your file manager. It is for installing by hand, reading the source, or carrying the connector to another computer — the install button above does the everyday job on its own.

On Linux: which Wine

MetaTrader 5 runs on Linux through Wine, and **MetaQuotes supports Wine 10.0 and above**. On anything older the terminal prints *"unstable and unsupported"* in its own log at every launch. This app reads that line and tells you which Wine your terminal is on, because a problem in the host underneath looks exactly like a problem in the app on top.

On Debian, Ubuntu and Mint the distribution's own Wine is usually too old. WineHQ's is not:

```
sudo apt install --install-recommends winehq-stable
```

> **Two traps when upgrading Wine.** Both are silent — neither gives an error you could search for.
>
> - **Close everything Wine is running first**, including processes you cannot see: `wineserver -k`. The old and new versions keep their sockets in different folders, so the new Wine cannot tell that the old one still holds your MetaTrader folder. It simply will not open, and says nothing.
> - **If you remove the old Wine packages**, the menu shortcut for MetaTrader may stop working: Debian and Ubuntu provide a `wine-stable` command that WineHQ does not, and the shortcut calls it by name. Clicking the icon then does nothing at all. Edit the `.desktop` file and change `wine-stable` to `wine`.

Installing MetaTrader 5 on Linux, or handing it to an AI

A smooth connection depends mostly on MetaTrader being set up right on Wine: Wine from WineHQ, a prefix of its own (`~/.wine-mt5`, which this app finds by itself) and that prefix in **Windows 10** mode — on Windows 7 it fails in ways that do not look like Wine. The verified procedure, step by step, with each step's check and its four traps, travels in the bridge folder as `INSTALL-MT5-ON-LINUX.txt`.

If you would rather hand it to an AI assistant with access to your terminal, give it `PROMPT-FOR-YOUR-AI.txt` from the same folder, or press **Copy the prompt for your AI** in the MetaTrader 5 panel when it finds no terminal. It carries the same steps and the limits it must keep: it asks before every `sudo`, removes nothing without asking, does not download the installer — you get it from your broker —, never touches your broker login or password, stops and asks at any unexpected output, and shows you the five checks at the end.

After an update

The program keeps the connector up to date by itself: when a new version brings a new connector, it replaces the old file in every MetaTrader where it finds one, as soon as it starts, and tells you. A running service keeps the code it started with, so MetaTrader goes on with the old one until it is closed and opened again; the panel says so, and how to switch without closing it: in MetaTrader, Navigator → Services → BotFactoryFX → Stop, then Start.

And when the connector does not answer, the panel reads MetaTrader's own files to tell you why: whether MetaTrader is closed or open, and whether the connector is missing, old, not set to start, or there twice. Whatever it is, the button next to it fixes it.

Connecting MetaTrader 4

MetaTrader 4 has its own bridge, this one's twin: it asks the same questions and answers the same things, and it speaks the same protocol, which is why it carries the same version number. What changes is what it is. MQL4 has no services, and the bridge is an **indicator** that sits on a chart, any chart: it draws nothing, and the symbol and the timeframe do not matter to it. An indicator and not an expert because, measured in a real MetaTrader 4, with the **AutoTrading** button off an expert never receives a single timer event while an indicator keeps running; and because MetaTrader 4 lets only experts and scripts trade: as an indicator, the bridge could not place an order even if someone wrote one into it. Nor does it take the chart's expert slot, so it can sit on the same chart as a robot.

The same button installs it, **Install the connector**, in the MetaTrader 4 panel of **Settings** → **Trading platforms**; it is also in **Live**, while no MetaTrader 4 is connected: **Connect the MetaTrader 4 where your robot runs** opens it. It copies the indicator into the terminal's `MQL4/Indicators` folder and, **with MetaTrader 4 closed**, adds to its profile an H1 chart that carries it, which MetaTrader 4 opens every time it starts. So close MetaTrader 4, press the button and open it again: on that chart, bottom left, a label starting with `BotFactoryFX Bridge` says the bridge is running and read-only, and the panel here turns green on its own.

With MetaTrader 4 open, the button only copies the file: when it closes, MetaTrader 4 rewrites its profile from what it has open, and a chart added from outside would be lost. The panel says so, and offers the way that does not wait: in MetaTrader 4's **Navigator**, drag **Indicators** → **BotFactoryFX** onto any chart. It starts at once, and MetaTrader 4 keeps it in the profile.

It talks in a folder of its own inside the common folder, not in the MetaTrader 5 bridge's: both MetaTraders share that folder on Windows, and also in a Wine prefix that has both, and each bridge would pick up the other one's questions. MetaTrader 4 keeps *orders*, not *deals*: each closed order arrives as its entry and its exit, and only closed ones. And it only hands over what its **Account History** tab shows: if only the last few months come through, right-click there and choose **All History**.

On an update, the program replaces the old indicator with the new one in every MetaTrader 4 where it finds one, without touching its profile; the one already running keeps its code until MetaTrader 4 is closed and opened again. To remove it, close the bridge's chart and delete the indicator from `MQL4/Indicators`. Its panel has the same three bridge-folder buttons, and the folder is saved in `~/.botfactoryfx/conectores/metatrader4` with its manual.

Connecting JForex

BFQuarantine reads the JForex account where your robot runs, and what it does there, through a small add-in that runs inside the platform. It reads the platform and **never places an order**.

How it connects

Through the same idea as MetaTrader 5: a small add-in called the **bridge** — *BotFactoryFX Bridge* — that runs inside the platform and talks to this app by writing files, here in `~/.botfactoryfx/jforex`. No port, no allow-list, no firewall rule. The difference is what the add-in is: JForex has no services, it runs *strategies* written in Java. The bridge travels in two forms: compiled, in a `.jfx`, which is what JForex starts, and as its source, `BotFactoryFX.java`, so you can read every line of it before you run it. The compiled one is built from that source with JForex's own compiler and its very options, and it is, byte for byte, the one JForex writes when it compiles it; it carries nothing of your account or your machine. It answers one request at a time, which is why the panel queues a second instrument instead of asking for both at once.

Installing it, once

On Linux and Windows this app installs it, whole and in one go:

1. Here, in **Settings** → **Trading platforms**, press **Install the bridge in JForex** in the JForex panel. The app finds your JForex — on Linux, through the shortcut its installer leaves or in `~/JForex4`; on Windows, through what its installer notes in the registry or in `C:\Program Files\JForex4` — puts the bridge in its strategies folder, compiled and with its source, and adds two lines to JForex's options (`JForex4.vmoptions`, with a copy of the previous ones beside it) so that it starts the bridge when it opens.
2. If JForex was open, close it. Open it and log in to your account: the bridge starts by itself, on your computer, a few seconds later (measured: 13 on Linux, about 20 on Windows), and the panel here turns green on its own. From then on it starts every time you open JForex, without you opening or starting anything inside it.

If your JForex already starts other strategies when it opens, that setting is left alone — yours would start too, and those may trade — and the panel tells you to start the bridge by hand.

On Windows, JForex installs itself in `C:\Program Files\JForex4`, which is also its data folder: it keeps its strategies there, and that is where this app writes. This was measured with JForex for Windows on Wine; whether

its installer lets a user who is not an administrator write in that folder is still to be measured on a real Windows. If it does not, the panel says so, with the path: give your user permission to write in that folder and press the button again.

On a Mac (where the app does not look for JForex yet, because it has not been measured there), or if it does not find yours, it is done by hand:

1. Open JForex and log in to your account.
2. Here, press **Copy the connector**: it leaves the bridge, compiled and with its source, in `~/.botfactoryfx/jforex`, and shows the path of the compiled one, with a button to copy it.
3. In JForex, on the **STRATEGIES** tab, press **Open** (the folder icon) and paste that path into the file name box: the folder's name starts with a dot, and on Linux and on a Mac the dialog does not show it.
4. Start it on your computer: **Start** → **Local Run** (JForex refuses a **Remote Run**, because the strategy asks for full access). It comes compiled, so it simply starts. Leave it running: the panel here turns green on its own.

The permission JForex asks for

The first time the bridge starts on an account, JForex shows a dialog — the **Full Access Disclaimer** — asking whether the strategy may have **full access**. Accept it with **I Agree**. Dukascopy keeps the answer on its server, with the account: on an account that has already accepted it, JForex asks nothing, not even on another computer. The strategy carries the annotation `@RequiresFullAccess` because JForex's policy only lets a strategy write inside its own files folder, and the bridge has to write its answers in `~/.botfactoryfx/jforex`, where this app reads them. The permission is about *where it may write*, not about trading: the bridge has no code that places, changes or cancels an order, and you can check that in its source. The manual shows the dialog.

The bridge folder

The panel also has three buttons: **Save the bridge folder (connector, manual, icon)** copies the whole folder — the compiled bridge and its source, the icon, a `LEEME.txt` and the manual as a PDF — to `~/.botfactoryfx/conectores/jforex` and shows the path; **Open the manual** saves it and opens the PDF; **Open the folder** saves it and opens that folder in your file manager.

After an update

A running strategy keeps the code it started with. So a new version of BFQuarantine brings a new bridge, but the one JForex is running is still the old one. If this app installed it, the panel, as soon as you open it, puts the new one in JForex's strategies folder by itself and tells you so: it takes over the next time you open JForex. By hand: copy it again, stop the strategy and start it again. The panel compares the version that is running with the one it ships, and tells you.

If something does not answer

- **No JForex platform is answering**: is the strategy running? Installed by this app, it starts when you open JForex and log in, not before: if JForex was already open when you installed it, close it and open it again. Its console line reads `BotFactoryFX Bridge … escribiendo en …`, and it writes a status file every few seconds; if the platform has been closed, the panel notices on its own.
- **Only your subscribed ones show up**: a bridge from an earlier version is running, which offered only those. Installed by this app, close JForex and open it again; by hand, copy it again and restart the strategy.
- **Entries in STRATEGIES you did not put there**, with paths from another computer: that list lives in your account's cloud, and anyone who uses the same account on two computers sees the other one's entries there. To load the bridge by hand, choose the file with **Open**, not an entry in the list: opened from the list, JForex compiles it in its cloud and leaves nothing on the disk. The automatic start does not use the list.

QUARANTINE AND LIVE

Quarantine

Quarantine holds the robot you were sold against the backtest it was sold with. Its code is never run: it is read for the numbers it declares. What is compared is its promise against the trades the account really made, so nothing is sent anywhere and nothing can be lost.

Putting one in

- **The robot** — its file (`.mq4`, `.mq5`, `.pine` or `.java`) and, if you have it, the backtest it was sold with: the Strategy Tester report, a list of trades in CSV, a PDF or a `.sqx`. That backtest is what its promise is measured against.
- **Broker or venue** — the account where it runs: a MetaTrader 5, a MetaTrader 4 or a JForex with its connector running (both are connected in *Settings → Trading platforms*, or from *Live*), or a Hyperliquid wallet you added in Settings.
- **Live instrument** and **candle** — the ones it trades on.
- **Which robot** — in a MetaTrader account, the magic numbers that have traded that instrument, each with its trades, and the trades made by hand. It proposes the robot's own if the account uses it, or the only robot there; with two or more, you choose. With *all trades on this instrument*, everything counts, whoever made it.
- **Count from** — *its first trade here*, offered when the account already has trades of that robot: what it did after the dates its backtest covers is as new to it as what it does tomorrow, so it counts from the start; the trades inside those dates are left out, because the robot had already seen that market. Or *today*, to start from zero.

Each time you open the section, what has arrived since is read and the record is brought up to date. Nothing runs in the background: the app does not need to be open for the robot to trade, only to pick up what happened.

The verdict looks at the money first

What matters is whether it makes the money its backtest promised, so the verdict compares the **profit factor** — what the winners made divided by what the losers lost— live against the one promised. With few trades a profit factor says little, so a band is drawn around it by resampling its own trades two thousand times: it only says *doing worse than its backtest promised* when the promise sits above that band with 90% confidence, and never before 20 trades. The hit rate is shown too, as information: a robot can win less often than promised and still make its money, or win as often and lose it. Five answers: no trades yet, too early to tell, losing money but not yet clearly worse than promised, making money as promised (or better), and doing worse than promised. Measured on 32 real robots built with StrategyQuant X against what they did in months they had never seen, this caught 12 of the 21 that lost money without one false alarm among the 11 that did not; the hit rate alone caught 10 and raised 4 false alarms. If the backtest gives no profit factor, the hit rate is compared as before, with a 95% Wilson interval.

Live accounts and wallets

The Live section reads the account where the robot runs and draws its trades on the candles they were made on: every entry and exit, the result, the hit rate, and what the market entries really cost against the close of the bar that signalled them.

- **MetaTrader 5** — through its connector, with no key and no open port. While no MetaTrader is connected, this section offers to install the connector, which sets it to start with MetaTrader: once MetaTrader opens, the

account shows up on its own. It reads *trades* rather than ledger entries: it pairs each entry with its exit and nets out commission and swap.

- **JForex** — through its own connector, a small strategy that only reads. While no JForex is connected, this section offers to install it: on Linux and Windows it puts it inside JForex, set to start with it, and once you open JForex and log in, the account shows up on its own; if it does not find your JForex, it copies it to your machine and gives the steps to start it by hand.
- **Hyperliquid** — from the wallet address: its closed trades here, and the account's curve on Home.

> **Reading an account needs no password and no key.** MetaTrader and JForex are read through files on this machine, and a Hyperliquid wallet by its public address. Nothing here can place an order.

Whether those trades still look like the backtest the robot was sold with is what *Quarantine* answers.

Wallets are added in Settings

An address belongs to whoever holds the key, not to a venue, so the list is not "the Hyperliquid account" but whatever wallets you like, named by you. Pasting an address tests it against the venues that can be read and fills the rest in. Hyperliquid shows two different addresses: your account's, in your account menu at the top right, and the one in its “Deposit” window, which is a mailbox that forwards the money to the account. It is the first one that goes here; with the second, the program warns that there is no account at it.

Frequently asked questions

Do I need to install anything?

No. The program is the whole thing, and what the quarantine records is kept on this machine. Reading a MetaTrader 5, MetaTrader 4 or JForex account needs its connector installed once: Live offers each one while none of it is connected.

Is the robot's code run?

No. It is read for the numbers it declares — the instrument, the lot, the stop and the target — and what is compared is its promise against the trades your account really made. Nothing is sent anywhere, and nothing here can place an order.

Which accounts can it read?

A MetaTrader 5, MetaTrader 4 or JForex account, each through its connector (installed from Settings → Trading platforms, or from Live), and a Hyperliquid wallet, by its public address (added in Settings). It never asks for a password or a key.

How do I scroll the chart back in time?

Drag it, hold **shift** and use the wheel, use the arrow buttons above the chart, or click the chart and use ← → (Page Up / Page Down for a full screen, Home and End for the two ends). The wheel on its own zooms rather than scrolls — that is the standard charting convention, and it is why the chart can look like it only zooms until you know about shift.

Metrics glossary

Profit Factor

`gross profit / gross loss`

> 1 = profitable, > 1.5 = good, > 2 = excellent. ∞ if no losses.

Win Rate

`winning trades / total × 100`

% of positive trades. Only useful with context.

Shortcuts and tips

- In Quarantine, load the robot's file and, if you have it, the backtest it was sold with: that report is what its promise is measured against.
- Ctrl+K searches the sections and this Help, and takes you to what you pick; accents and capitals do not matter.

The content of this app is educational. It does not constitute financial advice. Trading carries risk of loss.

If something doesn’t work, reply to your purchase email or write to support@botfactoryfx.com.