

cTrader cBot Engineering Proof



Owned proof: a read-only cTrader Automate cBot that evaluates one hypothetical order against symbol and account constraints. It never calls an order-placement, modification or cancellation method.

Verified on 12 August 2026: the managed Release build completed with zero errors and the combined preflight and state-boundary suite passed 22 of 22 automated tests. Spotware's official cTrader CLI 5.9.10 rebuilt the four production files with no CLI compiler errors or warnings and identified the package as Type=cBot with AccessRights=None.

Boundary: the package has not yet passed the authorized cTrader demo-runtime checklist. Pure policy tests and official package metadata are not represented as broker-runtime, production or execution evidence.

What the Proof Checks

Check	Deterministic behavior
Volume	Minimum, maximum and minimum-plus-step alignment
Market state	Market-hours status and symbol trading mode
Quotes and spread	Bid/ask validity and observed spread against a configured ceiling
Protection distance	Requested stop-loss and take-profit against pip-based broker minimums
Distance units	Unknown non-pip units return WARN instead of a guessed result
Environment	Account type, demo/live flag, broker, running mode and server time in structured logs
Session and expiry	Half-open and overnight windows plus exact N-bar pending-order expiry
Ownership	Manual or foreign-owner objects are ignored before any mutation can be recommended
Restart baseline	One explicit daily-risk authority; persisted/history disagreement fails closed

Safety contract

Control	Evidence
Access rights	AccessRights.None
Trading side effects	No place, modify or cancel call in the proof
External access	No network, file-system or clipboard access
Policy isolation	Boundary source imports no cTrader API and recommends actions only
Result labels	PASS, WARN or FAIL with explicit codes
Meaning of PASS	Only the checked preconditions passed at that instant

Build and test result

Verification	Result
Managed Release build	PASS - 0 errors; two unrelated missing host LIB path warnings recorded
Automated suite	PASS - 22 passed, 0 failed, 0 skipped
Official CLI package	PASS - 4 sources, Type=cBot, AccessRights=None, no CLI warnings
Package identity	15,116 bytes - SHA-256 ED2A71156092...F803073CE6D
Evidence verifier	PASS - 3 of 3 synthetic A/B/C workflows
Runtime acceptance	PENDING - authorized cTrader demo required

Acceptance Matrix and Delivery Method

ID	Input state	Expected result
A01	Valid aligned volume, open market, valid quotes and allowed spread	PASS unless a documented non-pip distance unit requires WARN
A02	Volume inside range but off the broker step	FAIL / VOLUME_STEP
A03	Closed market or restricted symbol mode	FAIL with the relevant market or mode code
A04	Spread exceeds configured ceiling	FAIL / SPREAD
A05	Pip-based stop or target below broker minimum	FAIL with the relevant distance code
A06	Broker exposes an unsupported distance unit	WARN / DISTANCE_UNIT; preserve raw values
A07	Pending order reaches its exact N-bar expiry	Recommend cancellation only when the owner key matches
A08	Manual or foreign-owner object is outside session	IGNORE / FOREIGN_OWNER; no mutation recommendation
A09	Persisted and history-derived daily baselines disagree	FAIL / BASELINE_MISMATCH; select no authoritative value

Runtime publication gate

Before this proof is used as public runtime evidence, the current package must be imported into an authorized demo account and run through three cases: valid open-market input, invalid volume step, and deterministic market/spread failure. The evidence must show NO_TRADE=true, a demo environment, expected check codes, repeatable restart behavior and no position or history transaction caused by the run. The pure boundary cases remain specification evidence until they are wired into a separately funded product scope.

How this supports client work

The proof demonstrates the engineering pattern used for paid work: freeze exact behavior, validate broker and symbol constraints before execution, separate warnings from failures, record deterministic evidence, and deliver editable C# source with explicit acceptance cases. It does not demonstrate a trading edge or promise returns.

Official references

cTrader compiling: <https://help.ctrader.com/ctrader-algo/documentation/all-algos/compiling/>

cTrader CLI: <https://help.ctrader.com/ctrader-algo/documentation/ctrader-cli/>

Symbol API: <https://help.ctrader.com/ctrader-algo/references/MarketData/Symbols/Symbol/>

Account API: <https://help.ctrader.com/ctrader-algo/references/Account/IAccount/>

Symbol trading mode: <https://help.ctrader.com/ctrader-algo/references/MarketData/Symbols/SymbolTradingMode/>