

VERSION 1.2.0 / 04 OCTOBER 2026

QuantDeveloper Studio

User manual

A practical guide to historical data, strategy backtesting, parameter optimization, chronological research and manual execution.

DATA

Import and connect

BACKTEST

Inspect assumptions

OPTIMIZE

Explore parameters

RESEARCH

Evaluate chronology

DOCUMENTATION PREVIEW

This guide describes implemented functionality. Windows customer distribution is in preparation while merchant fulfillment and the documented third-party redistribution requirements are completed. No software binaries are offered with this document. Use installation instructions only with an approved customer package.

CONTENTS

A guide to your workflow

Read chapters 1-3 to get started, then follow the sections relevant to your edition and data. The Windows and Ubuntu capabilities are different.

01 Choose the right edition	3
02 Install and activate	4
03 Set up your workspace	5
04 Import historical files	6
05 Download data through native APIs	7
06 Use news and other history without future information	8
07 Run a first historical backtest	9
08 Choose or edit a strategy	10
09 Understand fills, costs and statistics	11
10 Inspect, compare and preserve results	12
11 Explore parameters with optimization	13
12 Run historical walk-forward and out-of-sample research	14
13 Use Ubuntu Headless	15
14 Understand the execution workspace	17
15 Connect and manage an execution session	18
16 Troubleshooting and reference	19

Use the supplied synthetic sample or data you are entitled to use. Start with one instrument and inspect its fills before running a larger search. Provider accounts, data subscriptions and broker permissions are separate from the software license.

CHAPTER 01

Choose the right edition

QuantDeveloper Studio combines historical data preparation, reproducible strategy evaluation and parameter research. The Windows application provides the graphical workflow. Ubuntu Headless provides commands for supported bar-based research jobs.

Capability	Windows Studio	Ubuntu Headless
Graphical workspace, charts and Silver/Dark appearance	Yes	No desktop interface
Native J-Quants, FMP and NewsAPI.ai download dialog	Yes	No download command
Imported or transferred historical bars	Yes	Yes
Historical news, economic and Treasury attachments	Yes	Yes, through configured local files
Moving-average, news-filtered moving-average, mean-reversion and breakout examples	Yes	Yes
Native trade replay and trade-momentum example	Yes	Use explicitly derived bars instead
Custom legacy ATS/C# strategy source	Compatibility compiler	Unsupported
Grid, Random and Genetic optimization	Yes	Yes
Walk-forward, fixed validation and fixed OOS research	Yes	Yes
Manual simulator and supported broker adapters	Yes	No execution command

Windows runs historical strategies in a separate compatibility worker. Ubuntu uses a managed bar engine and accepts the supported portable strategy identities. Arbitrary Windows strategy source cannot be moved to Ubuntu for execution.

For your first session, use a supplied synthetic dataset or data you are entitled to use. Start with one instrument, a short interval and a small parameter search. Examine the resulting fills and assumptions before expanding the workload.

The functions available to you also depend on your signed license. Research, Optimization, Execution and Headless are separate technical entitlements. Consult the order confirmation for the features included in your purchase; the application does not infer an entitlement from a price plan.

CHAPTER 02

Install and activate

Windows requirements

The reviewed Windows package requires:

- A 64-bit Windows installation.
- The .NET 10 Desktop Runtime for x64.
- .NET Framework 4.8 for the 32-bit historical compatibility worker.
- The 32-bit Microsoft Jet 4.0 OLE DB provider used by the compatibility environment.
- A writable installation folder and sufficient free storage for source data, snapshots and results.

The customer application does not require the .NET SDK. The developer build instructions have additional prerequisites and are not the customer installation procedure. No minimum memory or processor specification has been established in the reviewed release documentation; workload size and concurrent workers affect resource use.

Extract the complete approved archive to a directory you can write to, such as `C:\QuantDeveloperStudio`. Keep its folder structure intact. Run `Start-QuantDeveloper-Studio.cmd`. The launcher creates a private workspace in your local application-data directory. An explicit `--workspace PATH` selects another workspace.

Ubuntu requirements

The headless bundle requires the .NET 10 Runtime. Its documented tested environment is Ubuntu 24.04 x64. The runtime is not bundled. From a terminal, `dotnet --list-runtimes` should include `Microsoft.NETCore.App 10.x`. See chapter 13 for commands.

Activate the machine

1. Open **Activation** in Windows Studio.
2. Choose **Save request** or **Copy request**.
3. Send that request through the activation channel supplied with your order.
4. When you receive the signed `.qdllicense` file, choose **Import license file**. You can also paste the complete signed license into the activation page.
5. Check the displayed status, features and expiry.

License verification works offline. Downloads and broker connections still need their own network access. Activation is bound to the operating-system identity of the machine. Moving to another PC, or reinstalling an operating system in a way that changes that identity, requires reissue. Windows and Ubuntu have distinct activation identities.

Renew a time-limited license

Import the new signed file supplied for your renewed period. Renewal is a manual file import; the application does not retrieve a new license automatically. Expiry is an exclusive UTC boundary: at the stated expiry instant, the previous license is expired. There is no post-expiry grace period.

New licensed research operations and order submission or replacement require a valid entitlement. Existing results, exports, data inspection, broker reconciliation and order cancellation remain available. Expiry does not automatically close positions or cancel working orders.

CHAPTER 03

Set up your workspace

A workspace holds your imported datasets, download manifests, presets, runs, optimization records, research records and logs. Choose a location with enough space for repeated experiments: a run preserves input snapshots as well as results.

Windows allows one application window per workspace. You can use different workspaces independently. Keep a project's inputs and experiments together, and give runs meaningful names that describe the strategy, instrument, period and purpose.

The principal workflow is:

- 1. Data:** import files or download supported provider data.
- 2. Backtest:** select prices, optional historical attachments, strategy and costs.
- 3. Strategies:** inspect or edit a Windows strategy.
- 4. Optimize:** declare and evaluate parameter ranges.
- 5. Research:** preview and run chronological training, validation and evaluation plans.
- 6. Results:** inspect, compare and export completed evaluations.
- 7. Execution:** manage a separate manual simulator or broker session.

The appearance control is the circle at the upper-right corner, divided by a vertical line. Click the left half for **Silver** or the right half for **Dark**. The change applies immediately and the preference is retained. A theme change preserves active work.

Use **Save preset** on Backtest to preserve a configuration for reuse. **Load preset** restores it for another evaluation. Check the selected data, dates, costs and parameters after loading: a meaningful strategy comparison requires knowing what changed.

During a running job, the application shows progress and elapsed time. **Cancel** requests cancellation. Completed work and diagnostic records remain in the workspace. Avoid moving input files or deleting workspace folders while a job is running.

A fresh customer package contains synthetic examples, not a seller's research history or populated brokerage configuration. If the catalog is empty, import a supplied sample or your own historical data. No external broker connects automatically at startup.

CHAPTER 04

Import historical files

Normalized CSV

Open **Data > Import OHLCV CSV**. Select a file, inspect the preview and confirm its symbol, currency and interval. The importer creates an owned normalized copy. Your original file is read-only input.

A time-bar file uses an interval in seconds:

```
DateTime,Open,High,Low,Close,Volume,OpenInt,BarSize
2024-01-02T09:00:00,100,102,99,101,5000,0,300
2024-01-02T09:05:00,101,103,100,102,5200,0,300
```

A daily file can use Date,Open,High,Low,Close,Volume. OpenInt is optional. Native trade CSV uses DateTime,Price,Size. Use invariant numeric formatting. Preserve the actual symbol; dotted symbols are supported.

The importer checks the whole file before publication. Missing essential fields, invalid OHLC bounds, out-of-order rows, overlapping bars and inconsistent metadata fail validation. Do not label a price series as adjusted or assign an exchange timezone unless you know those properties from its source.

CSV timestamps use their recorded clock without a suffix. For ordinary imported data, declaring a clock is not an automatic timezone conversion. Online downloads have explicit UTC metadata even though normalized CSV timestamps have no Z suffix. Mixing UTC data with ambiguous legacy clocks is rejected.

Vendor import

Choose **Import vendor data** for the supported formats below. Review the archive member, symbol, timezone, currency, aggregation choice and interval before importing.

Source	Supported inputs	Research conversion
TickData	Supported compressed trade/quote CSV	Eligible trade-price or bid/ask-midpoint bars
cTrader	Observed OHLCV CSV with declared interval; bid/ask tick CSV	Existing bars or midpoint-derived bars
Databento	CSV or DBN versions 1-3 for MBP-1, BBO-1s and BBO-1m	Eligible midpoint or trade-price aggregation under the format rules

Supported archive handling includes gzip, ZIP, nested ZIP and zstd. Select the exact member when an archive contains multiple candidates. Databento MBO, CMBP/CBBO, OHLCV and unsupported symbology modes are outside this importer's current scope.

Raw vendor events are preserved separately. They do not automatically become directly executable tick strategies: choose an eligible explicit bar conversion for the bar engine. cTrader bid/ask ticks are quotes, and a BBO record's carried-forward last price is not a new

trade. Midpoint bars provide research prices; they do not supply order-book fills or queue simulation.

CHAPTER 05

Download data through native APIs

Windows Studio includes native download support for J-Quants, Financial Modeling Prep and NewsAPI.ai/Event Registry. No separate Python download script is required.

1. Open **Data > Download API data**.
2. Select the provider and supported dataset kind.
3. Enter the provider API key. It is masked and held for the session rather than saved in presets, the catalog or download logs.
4. Supply symbols or an article query, source-date range, timezone and currency where applicable.
5. Review the request, page, record, byte and rate budgets.
6. Choose the relevant availability model for historical events, as explained in chapter 6.
7. Start the download and watch progress. After completion, inspect the dataset's coverage and quality notes.

The date range is inclusive in the requested source-date convention. The provider's entitlements, endpoint history limits and retention rules still apply. An API key does not guarantee access to every listed dataset.

Provider	Native download scope	Important distinction
J-Quants API v2	Daily OHLCV, minute OHLCV and trade ticks	Minute and tick access requires the provider's intraday add-on; tick downloads also produce minute companion bars
Financial Modeling Prep	Supported daily and one-minute prices; stock, forex and general news; economic calendar; Treasury yields	Only the implemented endpoint families are supported; no synthetic FMP trade ticks are created
NewsAPI.ai / Event Registry	Historical article search by keyword query, with optional language	Query-matched articles are attached as global news; stock-symbol associations are not invented

J-Quants tick bulk files cover the market before local symbol filtering, so a narrow symbol request can still require substantial transfer. The documented native tick retention is two years; provider availability may change. Missing minute intervals are not filled with invented prices or volume.

A completed download is published atomically through its manifest. Cancellation, malformed content or a limit that prevents complete retrieval does not publish a partial dataset as finished. Verified completed-cache reuse is available; changing or removing a cached payload makes verification fail.

Online prices are normalized to UTC. J-Quants daily bars cover a full Japanese calendar day converted to UTC. FMP daily research bars use fixed UTC calendar-day buckets, which are not

an exact exchange-session calendar. Inspect these conventions when choosing daily boundaries.

Preserved provider fields and metadata explain adjustment basis. The backtest engine does not automatically credit cash dividends or change position quantities for stock splits. Price history alone is not a complete corporate-action accounting model.

CHAPTER 06

Use news and other history without future information

Attach News, Economic or Treasury datasets separately from price datasets in **Backtest**. They provide information to a strategy; they are not executable market-price series. Attached history is retained when the specification is passed to optimization or research.

Historical information has several relevant times:

Time	Meaning
Event/publication time	The source's reported event or publication timestamp
Available time	The earliest time the selected model permits the strategy to see the record
Fetch time	When Studio observed the downloaded record

The strategy's history accessor returns only records whose available time is at or before the simulation clock. It returns defensive copies. Price data must be explicitly UTC when joined with auxiliary history; ambiguous clock combinations are rejected.

Choose an availability policy deliberately

Observed fetch, the default, makes a record available when it was downloaded. A historical article downloaded today is consequently invisible to a backtest ending before today. This is strict about what the current download proves, and a historical simulation may see no eligible news.

Historical publication plus delay models the archive as available at publication or release time plus a declared delay. It enables historical research, but it is an assumption about what could have been known then. Where supplied, a NewsAPI crawl timestamp provides an additional lower bound. An archive may contain later edits, revised sentiment or revised economic values rather than the original historical vintage.

Treasury observations often have a reference date without a verified release timestamp. The publication model uses the following UTC midnight plus the declared delay for those records. Do not treat a reference date as proof of intraday availability.

Use the news-filtered example

Choose **News-filtered moving average** and attach an eligible News dataset.

NewsBlockMinutes blocks new entries for the chosen number of minutes after available news. Existing positions can still exit. A value of zero disables this entry filter. It is an example of using point-in-time information, not a claim that news filtering improves performance.

Overlapping attachments that repeat the same provider/event/symbol identity are rejected. Choose a nonoverlapping set instead of counting an article twice. The shared history loader allows at most 32 attachments, 250,000 events and 256 MiB of auxiliary payload for a request.

Every evaluation preserves its selected history and provenance with hashes. That prevents later file replacement from silently changing an already defined experiment; it does not establish that the provider's archive is an unrevised original vintage.

CHAPTER 07

Run a first historical backtest

Use the short procedure below after importing a synthetic sample or a suitable historical price series.

1. Open **Backtest** and choose one compatible price dataset.
2. Check its symbol, currency, first/last timestamps, interval, clock convention and quality note.
3. Choose **Moving-average crossover** for a bar dataset. Start with valid FastPeriod and SlowPeriod values where FastPeriod is smaller.
4. Enter a start and end within the available coverage. The start is inclusive and the end is exclusive. Use the clock shown for the dataset.
5. Set initial capital, fixed order quantity, commission and slippage in basis points. Choose a quantity that makes sense for the instrument and account.
6. Attach historical events only if the selected strategy will use them, and check their availability policy.
7. Give the run a descriptive name, review the summary and select **Run backtest**.
8. Open the result and inspect both the equity curve and individual fills.

Start with a small period that contains enough observations for the indicator lengths. Indicators begin fresh at the run boundary; there is no hidden pre-period warm-up. A short run can legitimately produce no trades or unavailable statistics.

The service supports up to 32 selected market datasets with one price series per symbol. Selected instruments must share the supported account currency and unit-multiplier model. Begin with a single symbol before using a multi-instrument specification. The engine does not infer currency conversion or derivative contract multipliers.

For J-Quants native trades on Windows, select the raw Trade dataset and **Trade momentum**. Its signals and subsequent-trade fills use trade events. To use a bar strategy or Ubuntu Headless, select the separate minute companion. Those are different simulations and should have different run names.

One evaluation is limited to two million selected market observations. The Windows worker has a ten-minute per-run deadline. If a run fails, inspect the saved diagnostic record before increasing its size or starting a large search.

Optimization and research apply complete-bar cutoffs at phase boundaries. An ordinary backtest selects begins in its requested interval and may allow a selected bar to complete

after the nominal end. Check the saved execution and cutoff notes when comparing period boundaries.

CHAPTER 08

Choose or edit a strategy

The supplied strategies illustrate different signal rules and share explicit simulation assumptions.

Strategy	Input and purpose
Moving-average crossover	Completed-close fast and slow simple averages on bars
News-filtered moving average	Moving-average entries with an eligible-news cooldown
Mean reversion	A prior-window mean, population standard deviation and z-score rule
Channel breakout	Current close compared with prior high/low windows
Trade momentum	Windows native-trade example using trade-event signals

All supplied examples are long-only and use fixed units per instrument. Treat the defaults as demonstration settings. Verify that a parameter combination is valid and that the observation count is sufficient for it.

Windows source editing

From Backtest, choose **Inspect or edit source**, or open **Strategies**. Inspect a supplied template before modifying it. Compile, read the diagnostics, then run a short evaluation and inspect actual order behavior.

The Windows compatibility compiler supports **C# 5**. It expects a public, constructible `ATSCoordinate` implementation, including the supplied `EngineBarStrategy` templates. The desktop's .NET 10 runtime does not change the legacy strategy compiler's language level.

Parameters used by the application must be public writable numeric properties. Optimization attributes declare bounds and steps. Unsupported property types, unknown parameter names and invalid values are rejected. The precise source and parameter values are saved with a run.

Custom code is trusted code. The separate worker provides fault isolation, not a security sandbox. Review source from another party before compiling it. Research boundaries cannot prevent arbitrary custom code from reading an unrelated future-dated file or making its own network request.

Historical information in custom Windows code

The helper uses the current engine clock rather than accepting a caller-supplied future timestamp:

```
var news = StrategyHistory.Recent(Instrument.Symbol, 30, "News");  
var yieldValue = StrategyHistory.LatestValue("", "Treasury", "year10");
```

The first example asks for recent eligible news; the second requests the latest available global Treasury value. Handle absent data explicitly. Attaching a dataset does not mean its records are available at every simulated time.

Ubuntu runs the four exact supported portable bar strategies. Modified source, custom ATS classes and an unsupported strategy ID fail explicitly. There is no automatic migration of classic GUI projects into this workflow.

CHAPTER 09

Understand fills, costs and statistics

Historical execution model

Bar strategies act on completed bars. A market order fills at the next selected bar's open. An order created from the final available signal remains unfilled if there is no subsequent eligible fill event. The engine does not manufacture a terminal execution.

Windows native-trade replay evaluates trade signals and fills market orders on a subsequent trade. Same-timestamp events retain their numeric sequence. It does not treat a trade as a bid/ask quote or infer a queue position. Trade sizes beyond the compatibility worker's supported 32-bit size bound fail explicitly.

Both models assume the requested quantity can be filled. Slippage moves the execution price adversely by the configured basis points; commission applies to executed notional. One basis point is 0.01%. Neither model establishes available market capacity.

The supplied models do not enforce cash, margin or a leverage limit. An excessive fixed quantity can produce negative cash. They do not model a bid/ask spread, shared liquidity, market impact, queue priority or automatic currency conversion. Select inputs and sizing with these assumptions in mind.

Open positions are marked to the last observed price under the engine's normal model. There is no forced final liquidation. Corporate actions require particular care: the application does not independently credit dividends or adjust position quantities for splits.

Read the statistics

Result	Interpretation
Equity and net profit	Include realized results and the marked value of open positions
Return	Change relative to the evaluation's initial capital
Closed trades	FIFO matched lots, with proportional entry and exit fees
Maximum drawdown	Reported as a positive magnitude; chart drawdown observations are nonpositive
Sharpe	Observed daily closing-equity returns, sample standard deviation, zero risk-free rate and the saved annualization setting
CAGR	Uses elapsed time; unavailable below 365 days or with nonpositive final equity
Profit factor	Unavailable when no closed lot has a net loss

An unavailable statistic is not a zero, and a run without fills is not evidence of a successful strategy. Check fees, remaining positions and unfilled final orders before comparing net profit. The same strategy can differ materially under another clock, adjustment basis, cost model or fill convention.

CHAPTER 10

Inspect, compare and preserve results

Open a completed result to review account equity, drawdown, statistics, fills, closed lots and open positions. Use the chart to understand when changes occurred, then inspect the underlying records.

- Hover over the chart for values at an observation.
- Use the arrow keys for keyboard inspection.
- Zoom into a period and pan to examine adjacent activity.
- Double-click or choose **Reset view** to restore the full period.

Chart display sampling preserves extrema. It does not replace the full stored equity history. CSV exports retain every observation and full numeric precision.

Compare two completed runs when investigating a parameter or implementation change. Check that their source, price data, history policy, date range, sizing and costs match the intended experiment. A changed input is often more consequential than the displayed strategy name.

What a run saves

Artifact	Purpose
Original and resolved requests	Record what was submitted and the actual effective inputs
Strategy snapshot	Preserve the evaluated source or portable strategy identity
Input snapshots and provenance	Preserve selected price/history payloads and their metadata
Result JSON and CSV exports	Retain full metrics, equity, fills and trades
HTML report	Provide a portable view of the result
Environment and SHA-256 records	Identify data, engine, dependencies and evidence bytes
Logs and event records	Explain progress, failures and cancellation

Runs are stored below the workspace's runs directory. optimizations and research contain parent records that link their child evaluations. Presets and datasets are stored separately. Keep related files together when backing up or transferring an experiment.

A content hash detects a changed file; it does not prove that a vendor's data is correct or legally reusable. Preserve the original data entitlement, source notes and adjustment assumptions separately where needed.

Before sharing an archive or diagnostic record, review it for account identifiers, proprietary strategy source, licensed market data, article content and local paths. A full research workspace is not a ready-made public report. Prefer the specific report or export needed by the recipient.

CHAPTER 11

Explore parameters with optimization

Optimization evaluates a declared finite set of parameter choices. It saves failed and unranked trials as well as successful results.

1. Prepare and verify a Backtest specification.
2. Choose **Explore parameter ranges**, or open **Optimize**.
3. Enable the numeric parameters to explore. Set minimum, maximum and step; leave the others fixed.
4. Select Grid, Random or Genetic and a ranking objective.
5. Set the seed, trial budget and parallel workers.
6. Optionally reserve a final percentage for validation and choose how many leading training candidates to validate.
7. Review the configuration, start the search and inspect the retained trial diagnostics.

Method	What it does	Use it when
Grid	Evaluates the declared finite grid	The complete space fits the selected budget
Random	Samples finite-grid candidates without replacement using a seed	A bounded sample is useful before a wider study
Genetic	Evolves candidates on the same exact parameter-step grid	You want a budgeted adaptive search with saved generations

Objectives include net profit, return, Sharpe, low drawdown and profit factor. A trial with no fills or an undefined required score is unranked. The ranking uses training results. A high rank says how a candidate performed under this experiment's model; it does not establish future profitability.

The interface allows 1-16 parallel workers, up to eight enabled dimensions and 1,000 trials per experiment. More workers require more memory and input-copy activity. Start conservatively and monitor resource use. Grid rejects a space above the selected trial limit.

Genetic search uses finite-grid ordinals, tournament selection, crossover, mutation and elitism. It caches repeated candidates, including failed or unranked candidates. It can stop at its evaluation budget, generation limit, stagnation threshold or exhausted grid. Saved generations show candidates, fitness and the stop reason.

Reserved validation

You can reserve 0-50% of the final time range and validate the leading 1-20 training candidates. Each evaluation starts with fresh capital, positions and indicators. Bars crossing a phase's completion cutoff are excluded. Validation does not contribute to Genetic fitness or parent selection.

Cancel stops new scheduling and requests active evaluations to stop. Completed trials remain saved. A cancelled optimization is retained as cancelled, not presented as a completed search. Repeatedly adjusting the strategy after seeing validation results makes those observations part of your wider selection process.

CHAPTER 12

Run historical walk-forward and out-of-sample research

Research organizes historical evaluations into explicit chronological phases. It helps preserve what each selection was allowed to inspect.

1. Configure prices, history, strategy and execution assumptions in Backtest.
2. Configure the parameter search in Optimize.
3. Open **Research** and choose **Historical walk-forward**.
4. Select rolling or anchored training. **Suggest windows from these dates** creates an initial proposal; review it before use.
5. Set training, validation and evaluation durations, the selection rule and final OOS interval.
6. Preview the exact windows and complete run budget.
7. Run only after confirming that the selected data cover the intended phases.

Rolling training retains a fixed length and advances with each evaluation. Anchored training keeps its original beginning and grows. Historical evaluation windows are adjacent and disjoint. After development folds, a separate final calibration selects a candidate before the final out-of-sample evaluation.

Selection rule	Candidate choice	Role of validation
Training winner	Best eligible training candidate	Diagnostic; it cannot overrule that winner
Validation winner	Best validation result among declared leading training candidates	Part of selection

Evaluation and final OOS do not become fallback selection periods when an earlier phase fails. The saved record identifies the selected candidate, parameters, child runs and phase outcome.

Fixed validation and fixed OOS

Choose **Fixed-parameter validation** or **Fixed-parameter OOS** to evaluate one already declared parameter set over an explicit period. No search or candidate ranking runs for these modes. Preview the interval and run once. The result remains separate from walk-forward aggregates.

Interpret boundaries and aggregates correctly

Durations are fixed elapsed time in the retained dataset clock. A day is 24 hours; months and exchange sessions are not inferred. Bar begins are start-inclusive/end-exclusive, and research requires the entire bar to finish by the phase boundary.

Each training, validation and evaluation starts with fresh capital, positions and indicators. There is no hidden warm-up or carry between folds. Aggregate displays include sums of independent-fold net profit, means of individual returns and the worst individual-fold drawdown. They are not a continuously funded or compounded equity curve, stitched drawdown or combined Sharpe.

The final OOS designation applies to this saved experiment. Studio cannot determine whether you inspected the same dates in another study. Reusing OOS results to redesign a strategy makes them part of the broader research process. Preserve source, parameter choices, input snapshots and the reason for each experiment so that later review has context.

CHAPTER 13

Use Ubuntu Headless

Extract the Ubuntu bundle into a directory you own and install the .NET 10 Runtime. From that directory:

```
dotnet --list-runtimes
sh ./run-headless.sh help
sh ./run-headless.sh license request --output "$PWD/activation-request.json"
```

Send the request through your activation channel. Install the returned file and inspect status:

```
sh ./run-headless.sh license activate /path/to/signed-license.qdlicense
sh ./run-headless.sh license status
```

A single evaluation needs Headless and Research. Headless optimization and the research command require Headless, Research and Optimization. Import, catalog listing and validation can be used before activation.

Run the synthetic example

The supplied sample contains 240 generated hourly bars. It is software demonstration data.

```
mkdir -p "$PWD/demo"
sh ./run-headless.sh import \
  --file samples/headless/synthetic-bars.csv --symbol SYNTH --currency USD \
  --workspace "$PWD/demo/workspace" --output "$PWD/demo/dataset.json"

sh ./run-headless.sh validate --kind optimize \
  --config samples/headless/grid.json \
  --dataset "$PWD/demo/dataset.json" --workspace "$PWD/demo/workspace"

sh ./run-headless.sh optimize --config samples/headless/grid.json \
  --dataset "$PWD/demo/dataset.json" --workspace "$PWD/demo/workspace"
```

Use the supplied `random.json` or `genetic.json` for another search mode. run `--config samples/headless/backtest.json` evaluates one parameter set; `research --config samples/headless/research.json` runs the supplied walk-forward example. Include the same dataset and workspace options. Import/request output paths refuse to overwrite an existing file; use a new filename when repeating them.

Move a Windows-prepared dataset

Transfer the normalized price file, its metadata and all referenced history/provenance files. Set their paths to actual Ubuntu locations; absolute local paths avoid ambiguity in auxiliary metadata. The primary dataset `FilePath` can also be relative to the configuration directory. Windows paths are not automatically translated.

The CLI accepts transferred online bars and auxiliary history; it has no API-download command. Use a bar companion instead of native trade ticks. Use one of the four portable built-in strategies, not Windows custom source. Validate the complete configuration before starting a large job.

`--dataset` can repeat for distinct symbols and overrides the configuration's price-dataset selection. Unknown JSON fields and unknown parameter names fail. Numeric parameter values in the backtest specification are strings; optimization range values are numbers. Start from supplied configuration files for the actual schema.

Monitor and cancel

Append `--json` for newline-delimited progress and result events. The started event supplies the real job ID:

```
sh ./run-headless.sh status --id YOUR_JOB_ID --workspace /path/to/workspace
sh ./run-headless.sh cancel --id YOUR_JOB_ID --workspace /path/to/workspace
```

Ctrl+C, SIGINT and SIGTERM request cancellation. Completed work remains saved. Exit codes are 0 for success, 1 for a started job failure, 2 for invalid input, 3 for a licensing failure and 130 for cancellation.

CHAPTER 14

Understand the execution workspace

Execution is a separate manual workflow. A profitable backtest or selected optimization candidate does not automatically send orders to a broker. The application does not route historical strategy runs into live execution.

The initial adapter is an offline simulator and the session begins disarmed. No external connection is made at startup. Simulator fills require an explicit test-fill action; the simulator does not receive a live price feed or automatically match orders against a market.

The implemented broker scope is deliberately specific:

Adapter	Implemented scope	Setup and acceptance boundary
Simulator	Manual order review, submit, amendment, cancel and explicit partial/full synthetic fills	Offline; no real-money orders
IBKR TWS/Gateway	Documented account/contract discovery, supported orders, amendments, cancellation, fills, positions and reconciliation	Separate signed-in TWS/Gateway, API setup and account-specific acceptance
TT.NET	Documented account/instrument discovery, supported orders, fills, positions and reconciliation	Vendor license, app key, entitlements, exact environment and account acceptance
Kabu Station	Documented REST API 1.5 cash/margin-stock operations, cancellation and reconciliation	Local signed-in station, enabled API and current supported endpoint; no native amendment
Saxo OpenAPI	Whole-unit cash Stock/ETF, Market/Limit, permitted Day/GTC, native limit amendment and cancel	Registered app, account permissions and authenticated SIM/LIVE acceptance
Lightspeed	Documented order-message codec and state-reduction interface	Default authenticated connection is blocked pending the vendor session integration

The adapters have offline verification using controlled transports, callback fixtures or HTTP mocks. That verification is not an authenticated broker acceptance result or production certification. Confirm the adapter's supported product, order type, account and environment with the supplied release documentation before using it.

Saxo does not currently support CFDs, FX, futures, options, fractional quantities or stop orders in this adapter. It polls supported account/order information rather than streaming quotes. Kabu has no automatic cancel-and-replace substitute for an amendment: a new order follows an explicitly confirmed cancellation and a new review.

Broker fees, live market-data subscriptions, exchange access and vendor entitlements are separate from the software license. The execution workspace does not provide a universal portfolio-margin, FX conversion or liquidity model.

CHAPTER 15

Connect and manage an execution session

Practice the workflow with the simulator first. Choose a small synthetic order, review it, submit it, apply a partial test fill and observe the remaining quantity and position. Cancel the remainder and confirm the resulting state. These actions help establish how the interface represents orders and fills.

For an external session, the sequence is explicit:

1. Select the intended adapter and environment.
2. Complete that adapter's authentication or local application setup.
3. Select and verify the exact account.
4. Connect and reconcile the authoritative broker state.
5. Resolve the exact instrument and review its currency, lot size, price increments and supported order features.
6. Check quantity, notional, open-order and snapshot-age limits.
7. Arm the session, review the complete order and submit deliberately.
8. Monitor broker-confirmed status, fills and positions.

For IBKR, verify the actual account environment in TWS/Gateway. A socket port or account-name prefix alone is not authoritative proof of paper trading. Use a dedicated positive client ID and the documented API settings.

For TT, select the intended simulation, certification or production environment and exact account. For Kabu, distinguish its verification endpoint from a funded paper account; the local API does not return a complete authoritative account/environment identity. For Saxo, use the selected SIM or LIVE service and exact registered application/account permissions.

The coordinator saves order intent before transmission and reconciles uncertain outcomes. If a connection drops after submission, **do not assume the order failed**. A transport error can occur after the broker accepted it. The application does not blindly replay an unknown submission.

Wait for authoritative order, fill and position evidence, then follow the displayed reconciliation steps. Do not create a replacement order merely because the previous request timed out. Unknown outcomes remain unresolved until the available evidence can establish what happened.

Loss of a healthy session degrades or disarms execution. Ordinary connection does not preserve a previous armed state. Review and rearm only after the account and open orders have been reconciled. Submission and replacement require a currently valid Execution entitlement, including a check at transmission time.

Cancellation remains available while disarmed and after entitlement expiry. Disconnecting or closing Studio does **not** cancel working broker orders. Confirm their state in the broker's authoritative application when ending a session.

CHAPTER 16

Troubleshooting and reference

Use the checks below to identify the next step. Preserve the original error and related job record when asking for help.

Symptom	Check and next step
Windows launcher or worker will not start	Confirm all four Windows runtime prerequisites, intact extraction and a writable folder; inspect the reported error
Activation is invalid or expired	Check machine identity, signed file, features and UTC expiry; request the correct renewal or reissue
Dataset is visible but unavailable	Verify its normalized file exists and matches the recorded metadata/hash; reimport or restore the original verified payload
Download is not published	Inspect authentication, provider entitlement, date range and request/page/record/byte budgets; a failed or incomplete transfer is not a completed dataset
UTC/history attachment error	Use explicitly UTC price data; do not relabel ambiguous timestamps without a valid conversion
Historical news never blocks an entry	Check attachment type, symbol/global scope, available times, availability policy and NewsBlockMinutes
Overlapping history is rejected	Remove overlapping downloads or prepare a nonoverlapping set of events
No fills or unavailable metrics	Check indicator lengths, date coverage, valid parameters, next-event availability and sample length
A long run fails or times out	Reduce period/observation count, inspect diagnostics, and lower parallel workers for constrained machines
Ubuntu rejects a strategy	Select an exact supported portable built-in; Windows ATS/custom source is unsupported
A broker submission has an unknown outcome	Reconcile with authoritative broker state before any new order; a timeout is not proof of rejection
Lightspeed will not connect	The current default requires the missing authenticated vendor-session integration

Information useful for support

Record the product version, operating system, the action you attempted, exact error text and affected run/job ID. Include relevant diagnostics and a minimal reproducible synthetic example when possible. Review exports before sending them. Do not send API keys, passwords, broker tokens or signing material in a support report.

Preserve your workspace before replacing an installation. Keep the approved archive's notices and folder structure intact. Treat activation files as private customer material and follow the seller's replacement-machine procedure rather than copying another person's activation state.

Terms used in this guide

Term	Meaning
Bar	OHLC and volume for a declared time interval
Native trade	A price/size event replayed as a trade by the Windows worker
Auxiliary history	News, economic or Treasury information attached for strategy access
Available time	Earliest modeled time a historical event may be observed by the strategy
Snapshot	A preserved copy of the exact input used in an experiment
Validation	A declared evaluation used diagnostically or for selection, according to the chosen rule
OOS	A designated out-of-sample evaluation period for the saved experiment
Reconciliation	Comparing local intent/state with authoritative broker orders, fills and positions

About this edition

This guide describes version 1.2.0 and the native API data update reviewed on 4 October 2026. Provider availability, account permissions and future releases can differ.