QUANTRESEARCHER STUDIO

User manual

6 November 2026 · M219.1 public preview

From data import to strategy authoring, evaluation and result inspection.

https://quant-research.org/docs

Contents

M219.1: execution recovery and current workflows
1. Installation and first launch
1.1 Trial, subscription and PC transfers
2. A first backtest
3. Import and organize data
4. Write and save a strategy
5. Configure replay
6. Optimize and research
7. Inspect results
8. Connect a supported execution account
9. Keys, data and updates
10. Troubleshooting
Disclosures
M141 position selection and independent strategy instruments
Closing positions with working orders
Dataset defaults and search-range suggestions (M184)
Observed increments and dated JPX ticks (M188)

M219.1 edition · Updated 6 November 2026

QuantResearcher Studio brings local market-data preparation, C# strategy development, backtesting, parameter optimization, chronological research and supported broker execution into one workspace. Begin with the supplied synthetic data and a library example, then apply the same workflow to your own licensed data.

This manual describes the Windows desktop. Portable command-line components are separate from the desktop interface. Provider accounts, exchange permissions and available broker products depend on your own services. See Disclosures before interpreting results or connecting an account.

M219.1: execution recovery and current workflows

M219.1 builds on the M209 release with more responsive execution scheduling, clearer order timing, and improvements to strategy editing and optimization. Use the Windows installer or Ubuntu CLI/headless package for the same release. Bundled .NET and Databento DBN/Parquet reader dependencies require no separate setup.

Execution and pending orders

- **Submitted time** is the original submission time. **Last update time**, immediately beside it, records the last confirmed accepted submit or price amendment using the broker-provided time or exact response-receipt time. Polling, pending requests and rejected or unknown amendments do not advance this value. Older records without evidence remain blank.

- A temporary local sell-capacity conflict can wait for a newer account reconciliation and fresh quotes. Studio retains conservative reservations, retires a request proven not sent, and lets a later strategy decision produce a new request. It does not resend the old request. Journal records the position, reserved and requested quantities used by the decision and final check.

- A strategy waiting for fresh quotes can resume after fresh data and the required broker confirmation arrive. An amendment rejected locally before transmission does not permanently prevent subsequent quoting. **Stop** revokes automatic resumption. Existing working orders remain tracked; quote waiting alone does not cancel them.

- You can change the source or parameters of a **stopped** strategy while peer strategies run. A different strategy clears only the target's setup and trust; a compatible update retains its parameters and instruments. Editing a running instance still uses the existing validated-update handover or stop requirements.

- M219.1 avoids redundant full-account reads while the ordered command queue is busy. The reuse window is bounded by five seconds or half the configured hard account-age limit, whichever is shorter. Streamed account, position and fill changes invalidate that reuse. The original reconciliation age is retained; reuse does not pretend an old account snapshot is new.

Commands retain account-wide FIFO order, one request in flight, final risk/authorization/freshness/sell-capacity checks and uncertain-request reservations. Unknown and acknowledged-but-unreconciled requests are never resent. Network and serial broker acknowledgement delays can still produce a queue; offline regression tests do not guarantee live-broker latency.

Reading timing evidence

Record Quote.ReceivedUtc, the strategy's decision time, and the matching StrategyOrderUpdate.CommandId with Timing.SendStartedUtc, SendCompletedUtc and ResponseReceivedUtc. Provider timestamps, when supplied, help separate upstream delivery from host processing. Receipt-to-decision includes strategy gating; decision-to-send includes queued work and checks; send-to-response includes transport and broker response time. Missing timestamps remain null. **TIMING STUDIO_SCHEDULER** diagnostics additionally report poll gaps and time spent reading accounts, delivering events and dispatching commands. Compare synchronized clocks and the same request identity.

Optimization and large results

After **Load Opt. param**, explicitly choosing data for one instrument updates a preset's Symbol/SymbolName/InstrumentSymbol fixed value to that instrument. A symbol you type yourself stays intentional; multiple instruments are not guessed. Other fixed values, ranges, steps and candidate lists remain unchanged.

Backtest, Optimize and Research use compressed, verified worker responses for large outcomes, retaining fills, equity and position excursions. The decoded response budget is 512 MiB; ordinary metadata budgets remain bounded. This fixes the earlier 32 MiB worker-response failure without truncating evidence. It is not unlimited storage, and an older failed run with missing evidence must be rerun.

1. Installation and first launch

Use the signed Windows Setup from the Downloads page. It installs for the current Windows user and appears in Installed apps. Keep your workspace outside the application directory. The installer includes the .NET runtime, SDK/compiler and reference pack; a separate global SDK or Python installation is not needed for normal use or CSV imports. The strategy editor requires Microsoft Edge WebView2 Evergreen Runtime.

Ubuntu provides the CLI/headless edition. Install the verified AppImage using the supplied install_ubuntu.sh script to make the qr-studio command available. Ubuntu does not include the Windows desktop interface. See the Ubuntu command guide for data import, backtest, optimization and research commands.

For Windows updates, stop active tasks and strategies, disconnect sessions, then use **Settings → Update available · Install and restart**. On Ubuntu, use qr-studio update check and qr-studio update install. Downloads are checked against the publisher signature and hashes. Application updates and uninstalling preserve the separate workspace and activation; deactivate before uninstalling if moving to another PC. Strategies never start automatically after an update.

CSV, CSV.GZ and ZIP CSV imports use the bundled reader. Databento DBN and Parquet imports use the bundled isolated Python runtime and pinned dependencies; no manual Python setup is needed. Missing or damaged bundled files require repair/reinstallation.

On first launch, open **Settings** and choose your display time zone and base-currency default. Display settings change how timestamps appear; they do not rewrite stored timestamps or convert prices. The currency default initializes new research forms; it does not change a broker account's currency.

Only one application process can own the same workspace. Starting another copy for that workspace does not open an independent editing session.

1.1 Trial, subscription and PC transfers

In the installed Windows edition, open **Activation** → **Start 14-day trial**. On Ubuntu, run qr-studio license trial. The server starts a single 14-day period at the first successful activation, not at download or installation. No card is needed and the trial never creates a charge. A transfer does not restart the trial. The 14-day policy applies only to new trials started with M219.1 or later. Existing seven-day trials keep their original expiry; installing an update does not extend them. Older clients must update before starting a new trial. The Activation page shows the confirmed expiry in local time and UTC.

Online checkout is disabled while KOMOJU integration is prepared. Starting a trial never creates a payment or subscription. Keep the key from **Copy licence key** (Windows) or license key (Ubuntu) for recovery and PC transfers. Contact licensing support for purchasing information.

When online subscriptions become available, confirmed monthly payments will update the same licence automatically. Studio checks activation every 30 seconds while running; Ubuntu commands obtain online authorization when needed. Check **Activation** or license status for the entitlement expiry. The short online authorization lease is separate from the trial/paid expiry. Failed or pending payments do not extend access. Retain a delayed payment's reference and contact support instead of paying again.

One PC may be active across Windows and Ubuntu. Copy and keep the key **before** choosing **Deactivate this PC** or running license deactivate; the old PC clears its saved key after confirmation. Wait for the stated transfer time, then activate that key on the replacement PC. Support can reset a lost PC after ownership verification. If you already have a PayPal subscription, deactivation and uninstalling do not cancel it; cancel it in PayPal before its next billing date. Cancellation preserves already paid access through its expiry.

2. A first backtest

1. Open **Data** and choose **Install samples** to create the supplied synthetic datasets.

2. Open **Strategies** → **Strategy library**, select **MovingAverageCrossover**, and open its source.

3. Choose **Validate C#**. Review any diagnostics in Problems.

4. Choose **Use this code in backtest**, then choose the synthetic bar dataset. This example uses completed bar closes; quote-only data is not its input.

5. Review the selected symbol, parameters, account currency, cash, commission, execution feed and date range. Keep equity capture enabled to inspect the path of the run.

6. Choose **Run backtest**. Selection and validation alone do not run a strategy.

7. Open the completed run in **Results**. Inspect the equity chart, positions, pending orders, Trade History and Log together.

The moving-average example targets a long position when the fast average exceeds the slow average, and otherwise targets flat. It must receive sufficient observations to warm up. It does not automatically close positions merely because the historical file ends.

3. Import and organize data

Importing files

Choose **Data** → **Import data…**. Use Ctrl or Shift in the file picker to select several files. Supported local formats include CSV, CSV.GZ, ZIP CSV, Parquet and DBN.ZST. Recognized files can proceed directly to the batch import window. Information comes from headers, contents and explicit filename/path clues; the application does not silently invent missing contract settings.

Check the final symbol, accepted-row and dropped-row counts. Each file has an independent result, so a failed file does not undo files already imported. **Cancel import** stops outstanding work and keeps completed datasets. At completion, use **Close** to return to the catalogue.

Import concurrency is bounded by processor and available-memory estimates. Disk throughput, decoding and the number of selected files also affect speed. More workers do not guarantee a proportional reduction in time.

Automatic inference and its limits

CSV inference recognizes Trade, Quote and Bar columns and time-zone clues such as JST. An explicit offset or source declaration takes precedence over a computer default. Bar timestamps and intervals must have a consistent meaning.

Parquet identity columns can split a file into several symbols. Recognized J-Quants daily files use the exact Code, raw JPY prices and the appropriate Tokyo session boundaries. Adjustment columns remain in source evidence; adjusted values are not silently substituted for raw prices.

Databento headers provide schema, symbols, mappings and UTC timing. Supported schemas include trades, several top-of-book quote formats and supported OHLCV bars. Supported single-venue MBO files can be reconstructed into top-of-book quotes. This does not create deeper executable order-book history or guarantee support for every DBN schema. Product-specific currency, tick, quantity and multiplier settings must be known. A continuous futures series does not itself model contract rolls or their costs.

If a file lacks sufficient metadata or has contradictory mappings, correct the source or import configuration before retrying. An unsuccessful inference is different from an invalid market row.

Dropped rows

New desktop imports skip invalid market rows, such as crossed quotes, invalid numeric or OHLC values and decreasing timestamps, while retaining valid rows. Files with no valid rows are not published. Structural CSV, encoding, archive, missing-column and resource-limit errors remain file errors.

Choose **Inspect dropped rows…** after importing to see the source file, symbol, source row number, reason and original decoded values. **Save all dropped rows** exports the full JSON-lines report. The report is not limited to the small set of examples shown in a dataset manifest. Original source files remain untouched.

Folders and combined daily history

The **Folders** view groups data by vendor, schema and symbol. **Combine daily files by symbol** presents compatible, non-overlapping daily partitions as a single multi-day entry. The original partitions and their provenance are retained; this view does not create a new merged CSV file.

Turn consolidation off to inspect, select or edit one source partition. Overlapping imports or incompatible price, currency, schema, timing or instrument settings remain separate. A workflow selection expands a combined entry into its original datasets; the existing 256-dataset request limit still applies.

The catalogue search matches readable names, symbols, sources and metadata. **From** and **Through** accept years, months, dates and explicit-offset timestamps, or calendar selection. Catalogue dates filter datasets whose coverage overlaps the interval. To restrict an actual run, use that workflow's **Date From** and **Date To**.

Double-click a row to inspect its first stored records, or use **Open chart**. Combined charts span compatible daily partitions and maintain indicator history across those boundaries.

Loading and removal

Catalogue status reports metadata loading and retained-data maintenance. **Checking retained datasets** checks unloaded-data retention and bundled-sample protection; it is not reimporting every active source file. Large catalogues and slow storage can still take time. Replay, preview and chart operations verify the event data they consume.

On Data, **Ctrl+A selects the entire filtered list**, including off-screen rows. Bulk selection runs in small batches; Escape stops it and the selected count shows progress. Hidden workflow pages are prepared when opened.

Unload selected permanently removes the selected data from Studio after confirmation, in one operation. A compact batch record makes the removal durable before the catalogue updates. Studio reclaims its imported copies and replay files in the background; active readers finish first, and interrupted cleanup resumes on the next launch. Original source files outside Studio storage and saved results are kept. To use the data again, reimport its source. Bundled samples remain protected.

4. Write and save a strategy

Open **New strategy**, a library example or **Open source…**. All 17 supplied examples and templates use one strategy class deriving directly from Robot. A source does not need a second helper base class.

```
using QuantResearch.Engine;
using QuantResearch.Strategies;

public sealed class MyStrategy : Robot
{
    public override void OnQuote(ReplayContext context, Quote quote)
    {
        SimpleMovingAverage fast = Indicators.SimpleMovingAverage(quote.Bid, 20);
        SimpleMovingAverage slow = Indicators.SimpleMovingAverage(quote.Bid, 50);
        if (!fast.IsReady || !slow.IsReady) return;

        SetTargetPosition(quote.Symbol, fast.Close > slow.Close ? 1m : 0m);
    }
}
```

This quote example targets one unit long or flat. Each indicator retains its own history. Close is its current value; IsReady indicates a complete warm-up window. A period of 20 quotes means 20 observations, not 20 minutes. Call indicators on every intended observation, before returning because an order is pending. For completed-bar signals, use OnBarClosed and bar.Close.

The indicator family

The same syntax supports SMA, EMA, RSI, Bollinger Bands, MACD and ATR:

```
ExponentialMovingAverage ema = Indicators.ExponentialMovingAverage(quote.Bid, 20);
RelativeStrengthIndex rsi = Indicators.RelativeStrengthIndex(quote.Bid, 14);
```

EMA uses an initial arithmetic-mean seed. Wilder RSI 14 needs 14 price changes, so 15 prices; an entirely flat series returns 50. Update both calls before checking ema.IsReady and rsi.IsReady. Bollinger Bands exposes Middle/Upper/Lower, while MACD exposes Line/Signal/Histogram and is ready only after its signal warm-up; default 12/26/9 requires 34 prices. ATR accepts high, low and close, normally from completed bars. Consult the API reference for exact signatures, values and warm-up rules. No indicator reads future observations or loads a hidden pre-run history.

SetTargetPosition uses signed net quantity: positive is long, negative is short and zero is flat. It accounts for its pending target orders. Actual quantities use the instrument's declared units; one unit is not universally one lot. For custom order logic, use the order methods in the API reference.

A separate Symbol parameter is optional. quote.Symbol, trade.Symbol and bar.Symbol identify the current observation. Symbols lists selected execution instruments. An omitted-symbol order uses the current event's symbol; at startup it requires exactly one selected instrument. With several instruments, supply the intended symbol explicitly. Existing explicit Symbol parameters remain supported.

Use [Parameter] properties for editable settings. Validation discovers the strategy type from compiler metadata; when a file contains several strategy types, select the intended one. There is no need to type the fully qualified name into the former text field.

Save source… or **Ctrl+S** saves the opened file. A new source or library snapshot asks for a destination on its first save. Validation saves a separate library snapshot and does not replace a source save. If the file changed externally, the application preserves your editor text and avoids silently overwriting the other edit.

The editor restores its latest draft, including unsaved text, after restart. Recovered custom code requires validation before running. Use **Validate C#** again after source changes or an application/API update. Live compiler diagnostics help while editing but do not make changed code eligible to run.

Apply validated edits automatically

In each Execution strategy instance, enable **Apply validated updates automatically**. The setting is off by default and is saved for that instance. A successful **Validate C#** or **F5** updates every enabled instance using the same named strategy and C# type, across open Execution tabs. Saving a file or live editor diagnostics alone does not apply changes.

A running instance briefly pauses, cancels only its own pending orders, waits for broker confirmation, then starts the new version automatically. Positions, selected instruments, event-feed settings and compatible saved parameters remain. New parameters use their defaults. Private fields and indicators reinitialize. A stopped instance stays stopped; validating unchanged code does not restart it.

Failed compilation or incompatible parameter values leaves the previous running version unchanged. Unknown outcomes, rejected cancellations, cleanup timeout or startup failure prevent automatic resume. Review the instance status and Journal. **Cancel update · keep stopped** or disabling the option cancels a pending handover. Disconnecting, disarming, explicitly stopping or restarting Studio never restores a queued automatic resume. See automatic strategy updates.

Run mode and order queries

The host exposes read-only Boolean properties on ReplayContext. Exactly one is true for a normal Studio run; custom hosts that do not supply a mode report RunMode == StrategyRunMode.Unknown and all five flags are false.

Property	True when
context.IsPaper	Broker simulation/demo execution, including Lightspeed Paper
context.IsLive	Execution on a live broker account
context.IsBacktest	Standalone historical backtest
context.IsOptimize	Standalone optimization trial
context.IsResearch	Any research training, validation or out-of-sample trial

Optimization and research flags are exclusive; they do not also set IsBacktest. These values identify execution purpose, not the quote vendor, and are available from startup through termination callbacks and derived bar callbacks. The same replay flags apply to Windows and Ubuntu command-line workflows.

```
if (context.IsPaper) Print("Paper execution");
var orders = context.PendingOrders.Where(x => x.Label == "maker");
var positions = context.Positions.Where(x => x.Label == "maker");
var firstOrder = context.PendingOrders.FirstOrDefault(x => x.Label == "maker");
```

Where returns a collection. Use FirstOrDefault for one match and check for null. LINQ is imported automatically by Studio's strategy compiler; an explicit using System.Linq; remains valid and is needed when compiling with another tool that does not provide it. An omitted label is null; x.Label == "" compiles but does not match null. Use string.IsNullOrEmpty(x.Label) for unlabeled orders. Raw

context.PendingOrders records cancel through context.CancelOrder(order.Id); context.Positions exposes position handles. Broker pending snapshots contain this strategy instance's orders; account positions may include other owners' exposure.

Editor suggestions use context.PendingOrders and context.Positions, with label filtering, lambda members, run-mode flags, Symbol.TickSize, Symbol.TimeZone, Symbol.NormalizePrice(...), and observed context.Bars / context.Quotes history. Legacy TradingPositions and TradingPendingOrders remain compatible but are not offered as primary context suggestions. Suggestions are curated API assistance; Validate C# remains the compiler check.

5. Configure replay

Backtest, Optimize and Research each have **Choose data** and **Clear data**. The chooser supports folder selection and Ctrl/Shift multi-selection. **Ctrl+A selects only the rows currently on screen**. Confirm with **Use selected data** inside the chooser. Clear data resets only that workflow's data/feed selection.

Compatible daily partitions run as one chronological history with one strategy instance across file boundaries. Select Quote and Trade together with Ctrl-click, or accept either kind first and add the other later. The chooser keeps the previously accepted complementary kind for the same symbol; selecting a different symbol replaces the old inputs. **Selected inputs** shows the combined selection. Use **Remove** or **Clear selection** to remove inputs explicitly.

Quote + Trade automatically selects Trade as the passive-fill source; both streams reach the strategy. The mixed market-order quote option below controls market fills. **Advanced · order-fill source** lets you override this, and preserves a prior choice for an existing mixed selection. This uses the existing Trade matching model, not an exchange queue simulation. Other combinations may still require an explicit fill source. A strategy that requires quotes can save a Trade-only draft, but cannot run until its required data is present.

Date From and **Date To** limit the run. Blank fields use available selected history. Calendar buttons choose dates. Year, month and date inputs use the display time zone; Date To includes the complete entered period. A value containing a time denotes an exact instant, and a datetime end is exclusive. Explicit UTC offsets are honored. Ambiguous daylight-saving times need an explicit offset. Review the precise interval shown in the run summary.

Execution and costs

With both **Quote + Trade** data selected, new Backtest, Optimize and Research forms also enable **Quote + Trade: market orders use fresh Bid/Ask; passive limits use trades**. Keep Trade as the order-fill source. Market buys consume the observed Ask size and sells consume the Bid size, including a fresh cached quote during timer-driven exits. **Market-order quote maximum age** defaults to 1 second and is editable. Quote size is shared by orders until a new quote arrives, even when the general liquidity model is Unbounded. Missing/zero size, stale quotes and unavailable prices do not fill; market orders do not fall back to trade prices. Limit and stop orders still match later trades. A close uses the current replay time, not the cached quote's earlier time. FlexMM's session-close guard remains active when flattening cannot be simulated. This model cannot establish actual venue execution or queue priority. Older saved requests retain their original matching settings until this option is enabled.

Instrument conventions retain declared metadata and saved edits. Missing tick/quantity values are estimated from a bounded sample and labelled as estimates. If the exchange region is unavailable, a consistent regional source timezone (for example Asia/Tokyo on a numbered TickWrite export) is offered as an editable **source timestamp estimate**. An export timezone can differ from the exchange, so verify it. This estimate does not enable verified JPX price bands or infer sessions/holidays; UTC and numeric offsets alone leave the exchange region unspecified.

New forms enable **Fill market orders on the current quote**. On Quote execution, a market order submitted for the symbol of the current quote buys at Ask or sells at Bid. This setting does not apply to another symbol's quote, a cached quote, a completed bar, information events, or other order types. Trade and BarOpen execution and other order rules retain their documented later-observation timing. Older saved requests keep their original settings.

Review initial cash, commission per fill, liquidity and instrument conventions. Observed-size liquidity can leave partial fills and pending remainders; it does not establish real exchange queue position.

ResearchLedger accounting can permit negative equity. A successful replay status means the run completed, not that a strategy was profitable or a real broker would accept every simulated order.

The account currency follows a newly selected common market-data currency. Cash and commission numbers are unchanged when their denomination changes, so review both. Mixed currencies require an explicit account currency and historical FX observations covering the run.

Oldest allowed FX rate is the maximum age, in seconds, of an already observed historical conversion rate. Entering an age does not supply or download a rate. Future conversion observations cannot price earlier events, and today's daily rate is not a replacement for historical FX data. If all inputs use the account currency, no conversion is needed.

Run status distinguishes verification, preparation, replay and saving. During replay, percentage and processed/total events describe replay progress; they are not an estimated time remaining.

6. Optimize and research

Optimize compares parameter candidates using Grid or Genetic search. New forms default to Genetic. Choose which parameters vary, define their finite ranges and review evaluation, concurrency and seed settings. Search ranks successful candidates by terminal net profit. Failed candidates do not become winners. Equity/drawdown recording is enabled by default for new forms; loaded requests preserve their saved choice.

Research separates each fold into training, validation and out-of-sample (OOS) windows. Training runs the search. Validation compares a shortlist from training. One selected candidate then runs once on that fold's OOS window. OOS results do not feed back into candidate selection for that fold.

Selecting data can suggest up to three expanding folds. Review them against real coverage and the requested date range. **Suggest folds from selected data** replaces the current fold entries; manually edited folds are otherwise retained. Each row needs a Fold ID and ordered Train, Validation and OOS start/end values. Common formats such as 2026-10-01, 2026/10/1 and explicit-offset timestamps are accepted. A date-only end includes that local day.

Each segment starts with fresh strategy state and initial cash. No previous segment's positions, indicator warm-up or FX seed are silently carried into the next. The overall OOS profit is a sum of independent segment results, not a compounded portfolio curve.

When a field is invalid, it is highlighted and the first problem is brought into view. If selected symbols have no shared coverage, clear/reselect data or choose compatible intervals. Folds cannot be invented for nonexistent observations. Review sparse periods even when a range can be suggested.

After Backtest, Optimize or Research finishes, Results switches to newest first, highlights the completed run, and displays that same run's details and equity curve. Refreshing the list while reviewing saved results preserves your selection and sort order.

In **Choose data**, press **Enter** while the data table or search box has focus to apply the full selection and close the window. This is the same as **Use selected data**, including Quote + Trade selections and adding either kind after the other. An empty or invalid choice stays open for correction.

Results builds a bounded display cache for saved runs and reuses it across restarts. The first visit to older results builds their cache progressively; later history loads and curve switches avoid decoding every trial and fill again. Original receipts and evidence hashes are checked on reuse. Large Trade History records load when that tab is opened. Technical details prepares a bounded preview only when expanded; original result files remain unchanged.

Hover over **DD %** in the Results list or the **MAX DD %** metric to see when the maximum percentage drawdown occurred, in your selected display timezone. New runs retain the exact first maximum observation before chart downsampling. Older runs show a matching recorded observation when available; if it was not retained, the tooltip says so rather than estimating a time. The maximum percentage and maximum cash drawdown can occur at different times.

7. Inspect results

Select a saved run in **Results**. Drag the divider below the result list to give more space to either the list or details.

- **Equity & drawdown** shows the retained path and maximum drawdown. Chart downsampling does not discard the separately retained exact running statistics.

- **Positions** and **Pending orders** show the final state. End of history does not force liquidation or create a later fill.

- **Search trials** and **Research folds** expose the corresponding experiments and selected outcomes.

- **Trade History** groups recorded fills into continuous net positions, from opening until flat or reversal. Scaling and partial exits share one row. Entry/exit prices are quantity-weighted averages; Realized P&L is the sum of recorded closing-fill P&L before commission. Open positions are labelled Open with no invented close. The original fill ledger remains in the saved result.

- **Log** shows strategy output. A price printed at order submission is not necessarily a fill price; use Trade History or OnFill for executions.

Sharpe is computed from consecutive observed UTC-day simple returns, using their mean divided by sample standard deviation, a zero benchmark and no annualization. Hover over its card for the calculation or the reason for **N/A**. Insufficient daily returns, nonpositive daily closing equity, zero variance or missing historical statistics can make it undefined. A one-day run can have many fills and still lack enough daily returns.

Older saved runs may lack equity or fill capture. Rerun with the desired recording options to obtain those records; existing outcomes are not rewritten. To remove an unwanted saved run, right-click its result row and choose **Delete result…**. Confirmation identifies the run. Its saved request, outcome, receipt and retained chart/fill/log data are removed; source files and datasets remain.

8. Connect a supported execution account

The public connection guide currently covers Lightspeed Connect and the local simulator.

Connection	Scope
Lightspeed Connect	Supported U.S. stock orders, positions, executions and account balances. Choose the correct Paper/Live environment and account. Each API key supports one active API connection.
Corporate Kabu.com API · Planned	Planned corporate Kabu.com API integration (Mitsubishi UFJ eSmart Securities). Not available in the current Studio release. Connection and trading scope will be documented after implementation and verification.

The planned integration targets the corporate API service described by Mitsubishi UFJ eSmart Securities, separately from the personal kabu STATION API. Required agreements, permissions, supported products and availability will be confirmed before release. No connection procedure is available in this edition.

Connect and choose market data

1. Select the intended Execution account tab, environment and account; connect and reconcile.

2. Resolve and apply the exact instrument mappings. Lightspeed USD cash-stock mappings use the documented stock price-grid policy rather than a default tick of 1. The base tick is 0.01; prices below USD 1 use the applicable 0.0001 band. Review any explicit special mapping.

3. Open **Data**. Lightspeed Connect supplies no quote feed, so **Connected broker feed** is unavailable and Databento is preselected. Select an exact dataset identifier from the dropdown, such as EQUS.MINI, enter the API key and map the provider's raw symbols to the selected instruments.

4. Confirm your data access and fresh observations. Choosing a dataset does not purchase an entitlement. EQUS.MINI has no trading-status messages; fresh valid quotes may enable Lightspeed **Paper** under its specific policy. This exception does not enable Live trading without the required status evidence.

5. Choose a validated strategy, review instruments, parameters and code trust, then explicitly **Start**. Alternatively, review and submit a manual order. Connecting alone never submits an order.

A successful feed connection closes its setup window after the selected secure-save operation succeeds. Errors keep the window open. Reopening it displays the saved profile without automatically reconnecting.

DOM, charts, positions and history

Databento MBP-1 supplies top-of-book observations, not full market depth. DOM can draw empty grid rows using instrument price rules; quantities appear only where supplied. Databento trades build minute candles. Minute/hour charts can request up to seven calendar days of historical one-minute bars, displaying the latest 100, 300 or 500 aggregated bars (default 300). Historical entitlements and usage charges apply. **Refresh history** performs another bounded request; sparse periods stay missing and forming bars remain partial. Chart backfill does not automatically warm strategy indicators.

Lightspeed position rows use recorded execution quantities and entry times, allocated FIFO against closes. Older unrecorded inventory remains explicitly unattributed with unknown fields blank. Account net

quantity still governs orders; selecting a row does not instruct the broker to close a particular tax lot. Closed portions appear in **History** with gross instrument-currency PnL when entry and exit evidence is available; fees are separate. Fresh bid/ask observations can update estimated open PnL without changing quantities.

Lightspeed short selling is unsupported. Pending sells reserve available long shares, so a new sell cannot exceed the remaining capacity. A displayed Databento price is not proof that the Paper simulator will fill an order at that price. Check broker order and fill evidence.

Stop and recover

Stop this strategy stops callbacks and requests cancellation of that instance's pending orders, including earlier activations. It does not select manual or other-strategy orders. Restart waits for exact broker outcomes; an acknowledgement or absence from current orders alone is insufficient.

Prepared, Acknowledged and Unknown requests receive background broker-status checks while connected. Recovery preserves the original request and never resends an uncertain order or restarts a strategy. Other instruments and pages remain usable when their own readiness checks pass. Review **Orders** and **Journal** for progress and diagnostics. Explicit **Reconcile account** refreshes broker evidence.

The execution Log defaults to timestamp · Strategy name | message. Enable **Show diagnostic details** for instance IDs, activation IDs and output channels. Copy/save uses the selected view.

9. Keys, data and updates

FMP and NewsAPI.ai keys are stored independently using Windows current-user DPAPI, outside the research workspace and exported artifacts. Restored keys are masked. **Forget saved key** removes only the selected provider's saved key. Provider selection, closing the dialog or starting a download can save the entered key. Other broker credentials follow their own connection/profile behavior; Saxo browser sign-in session tokens are session-scoped.

The offline editor does not send source to a remote editor service. Strategy files and recovery drafts are local source text. Execute only code you trust: the worker provides execution/termination containment, not a security sandbox for hostile code.

Keep backups of your workspace and original source files. Manual update checking requires a configured publisher endpoint and verification key; an unconfigured check does not download an update. Package verification checks consistency against its manifest and is distinct from publisher authentication.

10. Troubleshooting

Symptom	What to check
Editor unavailable	Install the required WebView2 runtime and use a complete, matching package. Read the editor's diagnostic.
SDK/startup message	Compare the package README with installed SDK/runtime versions; launch through the current verified shortcut. Do not mix files from releases.
Empty catalogue during startup	Read the loading status and allow metadata indexing to finish. This does not itself mean data was deleted.
Import completes with dropped rows	Inspect the rejected rows and exact counts. Decide whether the accepted observations are suitable for your analysis.
Import needs attention	Resolve missing/ambiguous mapping, source timing or instrument metadata. Structural errors are not ordinary droppable market rows.
Strategy will not run	Validate the current source, review highlighted parameters, confirm selected data and execution feed, and check warm-up length.
FX maximum-age error	Match the account to genuinely same-currency data, or include covering historical FX rates and a valid maximum age. Do not relabel prices.
No fold suggestion	Check shared symbol coverage, selected Date From/To and sufficient observations; Clear data can remove an unintended earlier selection.
Sharpe is N/A	Hover over the metric for the specific reason; inspect the number of daily returns and closing equity.
Broker connection works but prices/orders fail	Check exact account/contract, environment, provider entitlements and broker diagnostic. Reference lookup is not price entitlement.
Reconciliation fault locks instrument editing	Stop any active strategy, use Retry reconciliation where offered, then review the returned broker error. A retry does not submit orders.

For support, include the application release, operation, complete sanitized diagnostic, source/strategy identity, chosen data and date interval, and whether the issue occurred in replay or broker execution. Exclude keys, passwords and raw account identifiers from shared screenshots and logs.

Disclosures

Backtests and educational strategies are research tools. Replay prices, liquidity, costs, corporate actions, information availability and FX assumptions determine the result. They do not establish profitability, exchange queue position or future broker fills. Snapshot and publication-approximation information datasets have different availability meanings; publication metadata alone is not proof of a historical point-in-time archive.

Provider data, market subscriptions and broker permissions remain governed by the services you use. Importing data or seeing an integration in the interface does not grant an entitlement. Included software notices describe bundled dependencies; optional provider SDKs/readers and customer data are separate. Consult the site's service and seller disclosures for commercial terms, rather than inferring them from this technical manual.

M141 position selection and independent strategy instruments

Lightspeed can resolve supported USD stock mappings from authenticated account positions and orders during reconciliation. Selecting a strategy instrument is no longer a prerequisite for closing an existing mapped position. Current account evidence, usable market data and order-recovery checks still apply.

In **Positions**, use Ctrl-click or Shift-click to select rows, then right-click **Close selected positions…**. Review the combined quantity for each instrument and submit once. Rows of the same net instrument generate one offsetting order; unselected quantities remain. A rejected or unknown outcome stops the remaining batch without retrying. Broker tax-lot allocation may differ from the displayed FIFO rows.

On Lightspeed, open the **Instruments** cell of another stopped strategy and use **Add stock instrument** while a peer runs. Apply its own selection; the peer's activation, mappings and orders remain. Existing mappings cannot be removed or changed while strategies run. The newly selected symbol must already have matching, current market data before its strategy starts; the instrument picker does not reconnect the data feed.

Strategy code can read observed history with context.Bars.Open.Last(10) or context.Quotes.Bid.Last(8). Use Count or TryLast when fewer observations may exist. Last(0) is current; up to 4,096 observations are retained separately for each instrument and activation. Completed OHLCV is under context.Bars.Closed; restart clears history. See the Strategy API guide and M141 details.

Closing positions with working orders

For a Lightspeed net-position close or reduction, the review lists the current same-direction working orders that will be cancelled first. Confirming the review authorizes those exact cancellations and the displayed closing quantity. Studio holds strategy callbacks while it cancels, obtains final broker cancellation evidence, and submits the close. A long close remains SELL and a short cover remains BUY; it never converts a long close to a short entry.

You do not need to stop the strategy to close a position. If an order update is acknowledged but not yet reconciled, Studio briefly waits for broker confirmation under the same lock as strategy callbacks. A price-only requote of the same reviewed limit-order IDs and quantities is checked automatically. The strategy's private variables and activation remain intact.

If the position, quantities, fills or reviewed order IDs change, a cancellation is rejected or uncertain, or confirmation does not arrive within the bounded wait, Studio sends no closing order. Check Orders and Journal and review the remaining position again after reconciliation. Requests and diagnostics stay in Journal; nothing is automatically resent. Other pending orders remain active. Running strategies may place new orders after the action, so stop them first if you want the position to remain closed. Exact Saxo position-ID closes retain their existing workflow.

Strategy history warm-up and error location

Quote and bar history starts empty at each strategy activation. Guard earlier values with TryLast(offset, out value) before requesting orders. Last(1500) needs 1,501 observations; the chart's loaded history is separate. Newly validated code includes source-line information: runtime errors show the callback, file, line, exception type and message in the execution error and strategy panel. Fix and validate the indicated code, reconcile, and explicitly restart; failure does not resend an order. Old validated code without line information must be validated again.

You can choose code for a stopped second strategy while a peer runs. Applied quote subscriptions are retained until all strategies stop; selecting the second strategy does not remove subscriptions or reset the running peer. Lightspeed allows additional verified contracts and quote selections during this period. Each running strategy still needs its own instruments.

Large local data imports

Local data imports have no fixed row-count limit. Files are processed incrementally; you can cancel an import without publishing a partial dataset. File-size, record-length and field checks still apply. After updating, retry only files marked Needs attention; previously imported datasets remain available.

cTrader minute-bar CSV files

Automatic import recognizes the cTrader export profile Timestamp,Open,High,Low,Close,Volume under a cTrader source folder. Timestamp identifies the bar opening; completed high/low/close/volume become available one interval later. For files named SYMBOL Minute_JST.csv, the importer recognizes one-minute bars in Asia/Tokyo and the price currency from known FX or precious-metal pairs (for example XAUUSD → USD). Review file can override these suggestions. Other schemas and conflicting source metadata still require review. Import files and Inspect dropped rows use software rendering, like the main Studio window.

Tick CSV files with missing sizes

Tick exports may name Bid Size and Ask Size in the header while omitting those values in their rows. Studio accepts these optional missing cells, keeps sizes unknown, and preserves any sizes supplied on later rows. Required timestamps and bid/ask prices are still checked. Reimport only the files marked Needs attention; the originals do not need editing. Dataset preview also uses software rendering.

Viewport selection and feed-history recovery (M168)

A retained-history buffer overrun during Lightspeed strategy execution can recover automatically after confirmed cancellation of affected strategy orders and fresh quotes. Positions and Journal history remain; private strategy state and observation history reinitialize. Stop cancels automatic restart. Unknown requests are never resent. See the conditions and bounds.

Large unload selections

Unload accepts more than 1,000 source partitions, with background progress and Cancel task. New unloads are permanent; workspace cleanup runs in the background. A combined-symbol row can include many partitions. See Large dataset selections.

Dataset defaults and search-range suggestions (M184)

Backtest, Optimize and Research automatically fill recognized instrument parameters, including Symbol / SymbolName, from a single selected market instrument. Tick size, quantity step, pip size and exchange time zone use explicit metadata or conventions remembered for that exact dataset selection. Verified increments take priority. M188 and later can show clearly labelled observed-price/size estimates when verified settings are unavailable; these estimates are not exchange rules. JPX dated price bands require the appropriate instrument classification and raw, unadjusted prices. With several instruments, choose the strategy symbol explicitly. Manual parameter edits and loaded parameter sets are kept. **Use dataset values** explicitly replaces recognized instrument fields.

Numeric search rows receive bounded starting ranges based on their declared constraints, current values and recognized meanings (for example coefficients, Tanh thresholds, periods or tick distances). These are editable search suggestions, not optimal trading values or estimates fitted to price history. A row's **Suggest** button replaces that row's range; **Suggest ranges** replaces checked numeric ranges. Exact candidate lists and manual ranges remain unchanged until explicitly replaced. No extra parameters are selected automatically. Research uses the same controls for its training search, with ordinary OOS validation afterward.

The live **Search trials** window keeps running trials first, then queued and completed trials. Click a column header to choose another sort, such as Score descending; rows automatically move when that value changes. Selection stays with the same candidate.

Observed increments and dated JPX ticks (M188)

Choosing Quote or Trade data fills missing price/quantity increments from a bounded background sample. These are labelled **estimates**, remain editable, and do not override declared metadata, saved conventions or manual changes. The sample is at most 2,048 records / 4 MiB per file, up to four source partitions per symbol. An observed price grid is not an official exchange tick or a proof of the trading lot. Exchange time is taken from declared/saved settings or explicit venue evidence; JST timestamp formatting alone does not identify an exchange.

For ordinary JPX shares, **Load JPX tick references…** imports date-specific FLEX BP table assignments plus issue-information evidence. The reference exporter is tools/export_jpx_tick_references.py; it requires the user's reviewed FlexDom/jpxnorm decoder and original FLEX reference archives. It checks the whole compressed BP and II files, sequence continuity, applicable dates, security type 01 and the 100-share unit. Keep these private reference exports in your own workspace. The application does not ship licensed market-data archives.

Backtest, Optimize and Research use the proposed order price and Tokyo trading date to choose table 01/03's band. They apply it to limit/stop validation, amendments, normalization and protection prices. Missing dates and unsupported tables stop replay. On a table change, an off-grid pending order is rejected without repricing or resubmission. The reviewed table window is 2021-01-01 through 2027-02-28; the announced March 2027 regime requires a reviewed update. ETFs and other security types are not covered by this ordinary-share reference importer.

Data must contain raw exchange prices. For metadata marked unspecified, explicitly confirm **These are raw, unadjusted JPX prices**; that confirmation is retained in the run configuration. Provider-adjusted data is rejected even with that confirmation. Imported payloads are not rewritten. Selected reference periods and source-evidence hashes form part of the run identity.

```
var tickAtPrice = Symbol.TickSizeAt(proposedPrice);
var buyPrice = Symbol.NormalizePrice(proposedPrice, OrderSide.Buy);
var nextAsk = Symbol.NormalizePrice(bestAsk, OrderSide.Sell, levelsAway: 1);
```

Symbol.TickSize follows the latest observed quote bid/trade/bar price under dated JPX replay. Before the first market observation it exposes a base increment; use TickSizeAt(price) or NormalizePrice for actual order prices and do not cache one tick across price bands. If the table changes within the selected history, price-specific startup calls must wait for a dated observation. Without a dated schedule, replay retains its configured fixed tick. Existing FlexMM uses NormalizePrice for its ladder and therefore follows the configured bands.

Official rules: JPX tick sizes, JPX trading units.

Initial cash and predefined optimization choices

Initial cash appears at the top of Backtest, Optimize and Research. All three pages share the same value, in the account currency configured in Backtest. Completed results remain unchanged. Predefined string parameters use dropdowns for fixed values. Checking a parameter initially includes all declared candidates; **Values…** lets you narrow the selection. Saved explicit subsets remain intact. FlexMM declares all Buy/Sell/Exit shapes, features 1-6, operators 1-3 and final functions. The selected search method and evaluation budget still limit how many combinations are evaluated.

Trade History places **Close price** immediately after **Fill price**, with quantity-weighted exit and entry prices for each position. Fill time is its first entry; Close time is its final exit. Both time columns fit their

timestamps. Reversal fills are allocated to the closing and new positions, including proportional commissions. DD remains the position's worst observed unrealized loss (MAE). Original fills and saved results are preserved; older results without position links require a rerun. Dividends, settlement and financing cash flows are not closing fills. Corporate actions can change quantities; the DD tooltip shows both entry and exit quantities.

The Results list shows **Time elapsed** (whole-run wall-clock duration in minutes, to two decimals) and **Symbol** immediately after Date. Quote and trade inputs for the same symbol appear once; multi-symbol runs list each distinct symbol. These values come from the saved run, including failed runs when timing is available.

Funding defaults, CAGR and round trips (M191)

Untouched initial cash defaults to **1000000 JPY equivalent**, shared across Backtest, Optimize and Research. Non-JPY currencies use an editable estimate from the ECB reference dated 30 October 2026; the date is shown beside the field. Custom or loaded cash is preserved. Unsupported currencies require an explicit amount. This reference is never used as historical replay FX. Ubuntu workflow requests continue to specify their account currency and initial cash explicitly.

The Results list shows **CAGR** immediately after Sharpe ratio: (final equity / initial cash)^(365.25 / observed days) - 1, as a percentage. It uses the displayed result's first and last observed market events, not execution time. Research uses the displayed final OOS fold rather than chaining independent folds. Short periods can produce extreme annualized rates. Zero initial capital, negative final equity, zero duration or nonfinite arithmetic produce no value. CAGR measures endpoint growth, not the equity path or future returns. **# of Trades** counts completed round trips; still-open positions are shown separately in Trade History. Strategy name uses content-sized width.

FX reference: European Central Bank.

Pain Index and smoothness (M192)

Results adds **Pain Idx** after CAGR, then **Smoothness**. Pain Idx is the time-weighted average percentage drawdown: the area below the running equity peak divided by elapsed simulation time. Lower is better. Studio holds each post-event drawdown until the next event, including nights/weekends; simultaneous events add no duration. It records the summary before chart downsampling.

Smoothness uses the **Zephyr K-ratio**: regression slope divided by the standard error of that slope, using cumulative equity at consecutive observed UTC-day closes. Higher positive values indicate steadier upward growth; negative values indicate decline. It is not annualized. Missing dates are not filled, and the last date may be partial. Compare similar observation windows. Intraday losses can disappear from daily closes; inspect Pain Idx, maximum drawdown and the curve too. Flat/perfect straight lines, fewer than three observed dates or nonpositive initial capital produce no finite K-ratio. Pain also needs positive elapsed time.

Backtest uses its own replay; Optimize uses the best successful candidate; Research uses the displayed final independent OOS fold. Older downsampled runs without the new summaries show an em dash: rerun to calculate these metrics. Complete older traces can be calculated exactly without rewriting saved files. Date width fits its timestamp; tooltips document definitions and coverage.

Metric reference: Zephyr StatFACTS, Pain Index and Zephyr K-ratio.

Missing settings in a batch import

CSV/CSV.GZ batch imports automatically use the saved **Settings → Display time zone** and **Base currency** for missing source time zone and price currency. There is no additional input or confirmation step. Explicit source values take precedence. The import window shows which defaults were used. For Japan exchange time and yen, save **Tokyo Standard Time / Asia/Tokyo** and **JPY** before import. Use **Data → Edit metadata** afterward to correct imported conventions: a source-timezone correction rebuilds UTC replay timestamps, while a currency correction changes the declaration without converting prices. Changing Settings alone does not alter existing datasets or results. Symbols come from the Symbol column when present; filenames are the fallback. CompanyInfo.asc is not required.

The import window also explains its worker limit. Studio reserves one CPU thread and 2 GiB, then budgets 3 GiB per worker, capped by the file count. A 12-thread PC with a 16 GiB memory budget therefore uses four workers.

On first launch, the detected PC time zone supplies an editable currency default: Japan → JPY, European time zones → EUR, elsewhere → USD. Choosing another time zone in Settings updates the proposed Base currency; you can edit it before saving. Existing saved currency choices are kept when reopening Studio. These are UI defaults, not verified instrument currencies.

Optimization parameter files and trial text (M206)

In **Optimize**, **Save Opt. param** saves a .qroptparams file containing fixed parameter values, checked search parameters, numeric Min/Max/Step and exact candidate lists. **Load Opt. param** restores these settings for a compatible validated strategy. Data, date windows, cash and search limits stay as currently configured. Unchecked drafts are kept; loading marks saved ranges as intentional so dataset suggestions do not replace them. Different source code with the same schema requires a compatibility review. Ordinary .qrparams files remain fixed-value parameter sets.

In the live **Search trials** popup, click a **Parameters** cell to select text. Use **Ctrl+A** to select its full parameter list and **Ctrl+C** (or the text context menu) to copy it. Text is read-only; copying does not alter the trial.

Execution's **Choose a validated strategy** picker shows **Description** in place of the C# type column. Hover to read the full description, or search by name/description. C# type searches remain supported; the exact validated type still identifies the strategy internally. Descriptions belong to the exact source revision.

Fresh-quote waiting and independent instrument application (M207)

A running strategy enters **Waiting for fresh quotes** when its market quote is missing, invalid or not executable under the existing freshness/trading-status rules. Ordinary market and order-lifecycle callbacks pause for that instance. A new valid observation and confirmed outstanding broker outcomes permit automatic resumption of the same activation; OnStart and private state are not reset. Waiting-period market callbacks are skipped, and an incomplete forming bar is discarded. Healthy instances on other instruments continue.

Broker reconciliation, explicit Stop and managed-clock cancellations remain available. Existing working orders remain tracked; quote waiting itself does not cancel them. Commands known not to have been sent are recorded as rejected, never queued for retry. Unknown outcomes remain blocked and are never resent. A lost/replaced feed session, account/authorization error or strategy error still follows its normal recovery rules. Stop revokes the automatic quote resume.

After reconnecting, each row keeps its saved instruments but displays a blank Instruments cell until explicitly applied. Applying one row leaves every other row unchanged.