Tick History Check

Description and User Guide for the MetaTrader 5 indicator

Product	Tick History Check
Source file	`TickHistoryCheck.mq5`
Version	1.0
Platform	MetaTrader 5, custom indicator drawn in the chart window (MQL5)
Purpose	Check whether the broker's tick history is complete, still downloading, or has been reset or rebuilt
Document date	28 October 2026 by ForexEAandMore

1. Overview

The Tick History Dashboard checks the tick history of the chart symbol, one trading day at a time, and shows the result in a panel on the chart. It tells you whether the data is complete, still downloading, or appears to have been reset or rebuilt by the broker. The indicator only reads data. It does not place, modify or close any orders.

Key features

- Requests tick history from the broker server for the last 1 to 30 days and counts the ticks per day.
- Compares every day with a **saved baseline** from earlier runs and flags a **DROP** when the count falls by more than a set percentage.
- Shows the first and last tick time of each day, the change versus the baseline, and a status per day.
- Dashboard panel with an overall verdict, connection status, a scrollable daily table, an event log, built-in help and hover tooltips on every element.
- Animated bar in the header corner while a test pass is running.
- Popup alert and Experts-tab message when a suspected reset is detected.
- Saves a full report to a text file, manually with a button or automatically after each pass.
- Light on resources: one day is processed per step and only failed days are retried.

2. Background: why tick history can change

MetaTrader 5 does not own the tick history of a symbol. The terminal downloads it from the broker's server and keeps a local copy. Whatever the server provides is what the terminal shows and what the Strategy Tester uses. This means the data can change without any action on your side. Typical causes are:

- The broker purges, rebuilds or replaces stored tick data, for example after a server migration, a change of liquidity provider or price feed, or a data clean-up.
- A symbol is renamed or a contract is rolled over, which creates a new history.
- Older ticks are rebuilt from M1 bars, which changes the tick count and the price sequence.

- The local cache is deleted or invalidated and the terminal downloads a different or shorter history than before.
- The broker keeps only a limited window of ticks, so older data cannot be downloaded again.

Because the terminal has no memory of what the data looked like before, a change can only be detected by comparing the data against something recorded earlier. This is what the indicator's baseline file is for.

3. How it works

1. On start the indicator builds a list of days: today and the previous days of the broker's server calendar, up to the number set in `InpDays`. Saturdays and Sundays are marked as weekend and skipped.
2. It loads the baseline file for the current server and symbol, if one exists.
3. At every check step (default every 3 seconds) it processes **one** day. It calls `CopyTicksRange()` for the whole server day. This call also makes the terminal request the ticks from the broker server.
4. If the call returns an error, the day is marked **syncing** and is retried on the next pass. Only failed days are retried, so days that are already complete are not downloaded again.
5. When ticks are returned, the indicator stores the count and the first and last tick times, then compares the count with the baseline.
6. If the count is lower than the baseline by more than the threshold, the day is flagged **DROP!**, written to the log, printed to the Experts tab and, if enabled, shown as a popup alert.
7. When all days are done, the baseline file is saved, a report is saved if auto-save is on, and the indicator waits for the re-check interval (default 30 minutes) before starting a full pass again.
8. When the broker's server day changes, the table rolls forward by itself.

Detection rules

Situation	Result
The server returns no data yet (error)	Status **syncing**. The day is retried automatically.
A weekday returns 0 ticks	Status **EMPTY**.
The count is equal to or higher than the baseline	Status **OK**. The baseline is raised to the new count.
The count is lower than baseline x (1 - threshold)	Status **DROP!**. Logged, printed and alerted. The baseline is set to the new, lower count.
A flagged day later returns a count within the threshold of the old reference	The flag is cleared and the log shows **recovered**.
No baseline exists for the day	Shown as **new**. The first count becomes the baseline.

4. Installation and quick start

1. Copy `TickHistoryDashboard.mq5` into the `MQL5\Indicators` folder. In MetaTrader 5 use File > Open Data Folder to find it.
2. Open the file in MetaEditor and press **F7** to compile. Fix any reported error before continuing.

3. In the terminal, open Navigator > Indicators and drag **TickHistoryDashboard** onto a chart, or double-click it.
4. Review the inputs (section 7) and press OK.
5. Wait for the first pass. While it runs, the header shows **TESTING** with a moving bar. When it shows **IDLE**, the table is filled and the result is final for this pass.

> **Before you rely on it:** the symbol should be in Market Watch and the terminal must be connected to the broker. Try the indicator on a demo account or a test chart first.

5. Reading the dashboard

The panel is 660 pixels wide and sits in the top-left corner of the chart by default. Hover over any element to see a short description.

Panel layout

Area	Contents
Header	Title, the **test animation** and the minimize button.
Button row	Re-check now, Reset baseline, Hide/Show help, Save report.
Status	Eight rows with the verdict and the state of the connection and test (section below).
Daily tick counts	Table with one row per day. Shows 7 rows at once; Newer / Older buttons scroll through the rest.
Event log	The last 6 events of the session, newest first.
How to read this	Short built-in help. Can be hidden to save space.

Status section

Row	Meaning
Verdict	Overall result: **All checked days OK**, **Downloading / syncing**, **Some weekdays have no ticks**, or **WARNING: possible tick reset detected**. Colored green, yellow, orange and red.
Symbol / server	The chart symbol and the broker server whose history is checked.
Tick sync (M1)	Whether the terminal has finished synchronizing M1 history with the server.
M1 history from	Date of the oldest M1 bar the server provides for this symbol.
Live feed	Age of the latest received quote. A large value means no live ticks (market closed or connection lost).
Pass	Time the last pass finished and when the next full check will run.
All days	Counts of all checked days by status, including days that are scrolled out of view.
Last alert	The most recent suspected reset: the day and the count before and after.

Daily table columns

Column	Meaning
Date	Broker server day and weekday. Server time, not your local time.
Ticks	Number of ticks the server returned for that day.
Saved base	The reference count from earlier runs (read from the baseline file).
Change	Ticks compared with the saved base, in percent. Today's count keeps growing, so only a fall matters. A fall beyond the threshold is shown in red.
First tick / Last tick	Server time of the first and last tick of the day.
Status	One of the values below.

Status values

Status	Meaning
OK	Ticks were found and there is no significant drop versus the saved base.
DROP!	The count fell by more than the threshold. The broker may have purged or rebuilt its ticks.
EMPTY	0 ticks on a weekday. Not downloaded yet, or the broker has no data for that day.
syncing	The server returned nothing yet. The check is retried automatically.
queued	The day has not been checked yet in the current pass.
weekend	Saturdays and Sundays are skipped.

> **About 22:59:59 as the last tick:** some instruments, for example gold (XAUUSD), have a daily break at the end of the server day. A last tick shortly before midnight and a first tick at 00:00 are normal for them and are not a sign of missing data.

Header animation

While a test pass is running, a moving bar and the progress are shown in the top-right of the header, for example `TESTING [..===.....] 3/5`. The first number is the days checked, the second the total. When the pass ends, the text changes to **IDLE - last test OK** in green, or **ALERT - tick drop found** in red. The animation remains visible when the panel is minimized.

6. Buttons and controls

Control	Action
Minimize (- / +)	Collapses the panel to its title bar and restores it.
Re-check now	Starts a new full pass immediately instead of waiting for the next scheduled check.
Reset baseline	Accepts the current counts as the new saved baseline, clears all DROP flags and the last alert, and saves the baseline file. Use it after you have investigated a drop and accept the new data.
Hide help / Show help	Collapses or expands the "How to read this" section. The panel height adjusts.

Control	Action
Save report	Writes a report file with a date and time in its name (section 8).
Newer / Older	Scroll the daily table by one page. Shown only when more days are checked than rows are visible.

7. Input parameters

Input	Default	Description
`InpDays`	30	Number of days to check, including today. Range 1 to 30.
`InpTimerSec`	3	Seconds between single-day checks. Higher values reduce load on the terminal and server.
`InpRecheckMin`	30	Minutes between full re-checks after a pass has finished.
`InpDropPct`	10.0	A day is flagged DROP when its count falls by more than this percentage versus the saved base.
`InpAlert`	true	Show a popup alert when a drop is detected. The message is always printed to the Experts tab.
`InpPanelX`	10	Panel position in pixels from the left edge of the chart.
`InpPanelY`	20	Panel position in pixels from the top edge of the chart.
`InpVisibleRows`	7	Number of table rows shown at once. Use Newer / Older to scroll through the rest.
`InpShowHelp`	true	Start with the help section open or closed.
`InpAutoSave`	false	When true, a "latest" report file is overwritten after every completed pass.

8. Files created

All files are written to the terminal's `MQL5\Files` folder (File > Open Data Folder > MQL5 > Files).

File	Description
`TickCheck_<server>_<symbol>.csv`	The baseline. One line per day: date and the highest tick count seen. Written after every pass and read at start. Only the days in the current window are kept.
`TickReport_<symbol>_<yyyymmdd_hhmmss>.txt`	A report created by the Save report button. Each click creates a new file.
`TickReport_<symbol>_latest.txt`	The same report, overwritten after each pass when `InpAutoSave` is on.

Report contents

- Header: generation time (local and server), symbol, server, M1 sync state, M1 history start date, drop threshold.

- Verdict, summary line and the last alert.
- The complete daily table for **all** checked days as comma-separated lines (date, day, ticks, saved base, change %, first tick, last tick, status). It can be opened in Excel.
- The full session event log with date and time. The panel shows 6 events; the report keeps up to 5,000.

When a report is saved with the button, the full path is also printed to the Experts tab.

9. Interpreting the results

Everything is OK

The server returned ticks for every weekday and none fell below the baseline. This confirms the counts are consistent with earlier runs. It does not prove that the individual ticks are identical, only that the amount of data has not dropped.

A DROP is reported

A drop means the number of ticks for a day is clearly lower than it was before. Work through these checks:

1. Did you or anyone delete the terminal's history or the `bases` folder, or change the account or server? A new server creates a new history.
2. Check the Journal tab for history synchronization or server messages around the time of the change.
3. Wait and press Re-check now. If the broker is still sending data, the count may recover and the flag clears by itself.
4. Compare with your own exported ticks or a second source if you have one.
5. If the change is confirmed and accepted, press Reset baseline so future comparisons start from the new data. If your tests depend on the old data, treat the new data as different and rerun them.

EMPTY or syncing for a long time

Possible reasons are a lost connection, a symbol that is not in Market Watch, or a broker that does not store ticks for that day. Check the Live feed row and the connection status at the bottom of the terminal.

A small change on today's row

Today's tick count grows through the day, so a positive change is expected and not a problem.

10. Technical notes

- **Timer design.** A 200 ms timer drives the animation. Day checks are throttled to the `InpTimerSec` interval, so the animation stays smooth without increasing the load.
- **One day per step.** `CopyTicksRange()` loads a full day into memory and the array is freed right after counting. This avoids freezing the terminal and limits memory use on busy symbols.
- **Server time.** Day boundaries are in the broker's server time. Use the first and last tick times to relate them to your local time.
- **Interface objects.** The panel is built from chart objects with the prefix `THC_` and is removed when the indicator is removed. Text is limited to 63 characters per object, so longer text is shortened.

- **Baseline logic.** The baseline is the highest count seen for a day. On a drop it is set to the new lower count, so the same event alerts only once. The old value stays visible in the Saved base column until the flag clears or the baseline is reset.
- **No trading.** The indicator uses only read functions and file functions. It makes no trade requests.

11. Limitations

- The first run has no baseline, so nothing can be reported as a drop until a second run or later pass has data to compare against. Changes that happened before the first run cannot be detected.
- Weekends are skipped. Instruments that trade on Saturday or Sunday, for example some crypto symbols, are not checked on those days.
- Only the chart symbol is checked. Attach the indicator to each symbol you need to monitor.
- A count comparison detects missing or removed data. It cannot detect changed prices with the same number of ticks.
- The indicator shows what the terminal receives. It does not change the data used by the Strategy Tester, which has its own tick source.
- The panel needs about 540 pixels of chart height with the help section hidden and more with it shown. On a small chart, lower `InpVisibleRows`, hide the help or minimize the panel.
- The code has been written to be compiled and used in MetaEditor, but you should test it on your own terminal and broker before depending on it.

12. Troubleshooting

Symptom	Likely cause	Action
The panel is not visible or is cut off	The chart window is smaller than the panel.	Maximize the chart, change `InpPanelX` and `InpPanelY`, lower `InpVisibleRows`, hide the help or minimize the panel.
Days stay on syncing	No connection, symbol not in Market Watch, or the broker has no ticks for that day.	Check the connection and Market Watch. Press Re-check now later.
All weekdays show EMPTY	The broker does not provide tick history for this symbol.	Check with the broker or try a different symbol.
No popup alert	`InpAlert` is off or alerts are disabled in the terminal.	Enable `InpAlert`. The message is always in the Experts tab.
A report file cannot be found	It is stored in the terminal's data folder.	Use File > Open Data Folder > MQL5 > Files. The Experts tab shows the full path.
Panel text looks cut off	MetaTrader limits object text to 63 characters.	Hover for the full description. This is a display limit only.
Drop flags remain after you accepted the new data	The baseline still holds the old reference.	Press Reset baseline.
The compile fails	Source file changed or incomplete.	Use the supplied file unchanged. Compile with F7 and read the error list in MetaEditor.

13. Version history

Version	Changes
1.00	First indicator. Chart comment with per-day tick counts, baseline file and drop detection.
2.00	Full dashboard panel with verdict, status section, scrollable daily table, event log and built-in help. Added first and last tick times, change versus baseline, minimize and help toggle, Re-check and Reset baseline buttons, save report (manual and auto), header animation while testing, hover tooltips on every element, day roll-over and retry of failed days only.