

Trade Assistent V2.008

Funktionsprüfung gegen Anwender-Anforderungen

Analyse-Datum: 30.12.2025

Grundlage: statische Analyse der bereitgestellten Quelltexte (keine Laufzeittests im Terminal).

1. Analysierte Dateien

- Trade Assistent V2.008.mq5
- event.mqh
- gui_elemente.mqh
- trades_panel.mqh
- discord_send.mqh
- discord.mqh
- db_state.mqh
- ui_registry.mqh

2. Architekturüberblick

Haupt-Flow:

- **OnInit()** initialisiert Discord, DB, stellt Trade-State wieder her und baut UI neu auf.
- **OnChartEvent()** verarbeitet UI-Interaktionen (Button-Drag, Send-Click, Positions-Buttons, Line-Drag).
- **OnTick()** überwacht u. a. SL-Erreichung.
- **OnDeinit()** löscht registrierte Chart-Objekte.

Persistenz (SQLite): *meta* (key/value) + *positions* (symbol/tf/direction/trade_no/pos_no + entry/sl + sabio + status + flags).

Dateiname pro Symbol/Timeframe: **DowHowState_<Symbol>_<TF>.sqlite**.

3. Ergebnisübersicht

Erfüllt: 10 Teilweise: 6 Fehlend: 4

4. Prüfmatrix (Anforderungen 1–20)

Nr	Anforderung (Kurz)	Status	Code-Stellen
1	EA-Start: rechts 2 Linien (Entry/SL), 2 Buttons (Entry/SL), Send-Button, Sabio-Edits, Trend/Pos Eingaben	ERFÜLLT	OnInit() (Trade Assistent V2.008.mq5 Z.211), createEntryAndSLLinien() (Z.1140), SendButton()/SabioEdit() (gui_elemente.mqh Z.459/527)
2	Entry/SL-Button verschiebbar: andere Objekte wandern mit; Text zeigt Preis & Lot; Buy↔Sell wenn SL>Entry	ERFÜLLT	OnChartEvent() (event.mqh Z.86), movingState_R3/R5 (ab Z.233); Richtungslogik: if SL>Entry (um Z.282)
3	Entry/SL-Linien verschiebbar und lösen dieselbe Synchronisierung aus wie Buttons	TEILWEISE	OnChartEvent(): PR_HL/SL_HL Change (event.mqh Z.175) speichert DB, aber kein Code gefunden der Buttons/Edits an die Linienposition anpasst.
4	Links: Panel mit LONG/SHORT Label	ERFÜLLT	UI_TradesPanel_Create() (trades_panel.mqh Z.145; Aufruf in OnInit)
5	Active-Trade Buttons LONG/SHORT unsichtbar; sobald Trade existiert sichtbar (rot)	ERFÜLLT	showActive_long/short() (trades_panel.mqh ab Z.211); gesteuert in UI_TradesPanel_RebuildRows() (Z.485)
6	Cancel-Trade Buttons LONG/SHORT unsichtbar bis Trade existiert	ERFÜLLT	showCancel_long/short() (trades_panel.mqh ab Z.191); gesteuert in UI_TradesPanel_RebuildRows()

7	Pro Position: Tabellenzeile mit Infos + Cancel-Button + 'SL erreicht'-Button; zusätzlich Send-Button erzeugt Trade und sendet Discord	ERFÜLLT	Rows: UI_TradesPanel_RebuildRows() (trades_panel.mqh Z.485). Send: SENDTRADEBTN State-Check (event.mqh Z.382) → DiscordSend() (discord_send.mqh Z.14)
8	Cancel/Stop-Buttons: löschen Trade/Position, senden Discord; pro Trade Entry/SL-Linien mit Label, verschiebbar; Verschieben sendet Discord	ERFÜLLT	Panel-Handler: UI_TradesPanel_OnChartEvent() (trades_panel.mqh Z.721) → UI_CloseOnePositionAndNotify() (event.mqh Z.456). Linien+Label: CreateEntryAndSLLines() (gui_elemente.mqh Z.211) + UI_CreateOrUpdateLineTag() (Z.1265) + TP_FinalizeLineMove() (event.mqh Z.25)
9	Infos der Linien/Trades/Positionen pro Symbol werden in DB gespeichert	ERFÜLLT	db_state.mqh: DB_DefaultFile() (ab Z.54) + Schema (ab Z.161); Speichern Linien: DB_SaveLinePrices() (gui_elemente.mqh Z.101), Speichern Positionen: DB_UpsertPosition() (db_state.mqh Z.309)
10	Bei EA-Neuladen: aus DB lesen und Zustand wiederherstellen (Linien, Buttons, Panel, etc.)	TEILWEISE	DB_RestoreAll() (Trade Assistent V2.008.mq5 Z.284) + UI_TradesPanel_RebuildRows() stellen Trades/Positionen/Trade-Linien wieder her. Basis-UI (Entry/SL Buttons/Edits) wird aktuell neu zentriert erzeugt (siehe Punkt 16).
11	Anwender kann Trade-Nummer und Pos-Nummer überschreiben; EA nutzt diese fortlaufend	TEILWEISE	TRNB wird gelesen (DiscordSend() discord_send.mqh Z.14). POSNB wird aktuell nicht als pos_no verwendet (pos_no kommt aus DB_GetNextPosNo-Logik) und POSNB wird nach Send automatisch überschrieben.
12	Sabio-Preise: Eingabe in Editfeldern; Werte werden gespeichert und an Discord mitgesendet	TEILWEISE	Discord: DiscordSend() liest SabioEntry/SabioSL und speichert in DB_PositionRow. Persistenz: ENEDIT ruft UpdateSabioTP() (event.mqh Z.408), aber UpdateSabioTP() (Trade Assistent V2.008.mq5 Z.566) schreibt nichts in DB und konvertiert Strings nicht korrekt.
13	Buttons im Panel lösen passende Aktion aus (Grafikobjekte löschen, Trade/Pos in DB schließen, Discord senden)	ERFÜLLT	UI_TradesPanel_OnChartEvent() (trades_panel.mqh Z.721); DB_UpdatePositionStatus() (db_state.mqh Z.799)
14	Trades-Panel verschiebbar; Objekte wandern mit	FEHLT	Keine Drag-/Move-Implementierung für TP_BG/Panel-Position in event.mqh oder trades_panel.mqh gefunden (TP_X/TP_Y wirken statisch).
15	Tabellenliste aller Trades aus allen DB-Files (Excel-Export) – noch zu bauen	FEHLT	Keine Export-/Scan-Funktion über mehrere DB-Dateien gefunden.
16	Beim Neuladen: Entry/SL Buttons/Linien + Sabio-Edits sollen aus DB restauriert werden (nicht in Chart-Mitte erscheinen)	FEHLT	createEntryAndSLLinien() setzt Koordinaten aus Chart-Mitte & Preise aus iClose(); gespeicherte Meta-Werte (price_entry/price_sl) werden beim Init nicht angewendet.
17	Bei Verschieben von SL/Entry Button oder Linie muss sichtbarer Preis korrekt aktualisiert sein	TEILWEISE	Button-Drag: Preis-/Textupdate vorhanden (event.mqh movingState_R3/R5). Linien-Drag PR_HL/SL_HL: kein UI-Update für Buttons/Edits gefunden (nur DB_SaveLinePrices).
18	Wenn 'Send & Trade' aktiv: echte Broker-Position senden und in DB markieren	FEHLT	Broker-Trade-Code ist im Hauptfile auskommentiert (CTrade/BuyStop/SellStop). Click-Handler ruft nur DiscordSend().
19	SL erreicht: Discord-Meldung automatisch senden	ERFÜLLT	TPSLReached() (Trade Assistent V2.008.mq5 Z.602); Aufruf in OnTick() (Z.337)

20	Nach Load: offene Positionen prüfen; wenn SL erreicht -> Alert + Möglichkeit, dies noch an Discord zu senden	TEILWEISE	TPSLReached() läuft per OnTick und sendet i.d.R. sofort Discord + Alert. Eine bestätigungsbasierte Option („erst Alert, dann optional Discord“) ist nicht implementiert; OnInit prüft nicht aktiv vor dem ersten Tick.
----	--	-----------	--

5. Zentrale Funktionen und wo sie liegen

Funktion	Datei / Position
OnInit	Trade Assistent V2.008.mq5 (Zeile 211)
OnChartEvent	event.mqh (Zeile 86)
OnTick	Trade Assistent V2.008.mq5 (Zeile 337)
TPSLReached	Trade Assistent V2.008.mq5 (Zeile 602)
DiscordSend	discord_send.mqh (Zeile 14)
SendDiscordMessage	discord.mqh (Zeile 235)
DB_Init	db_state.mqh (Zeile 191)
DB_RestoreAll	Trade Assistent V2.008.mq5 (Zeile 284)
DB_SaveLinePrices	gui_elemente.mqh (Zeile 101)
UI_TradesPanel_Create	trades_panel.mqh (Zeile 145)
UI_TradesPanel_RebuildRows	trades_panel.mqh (Zeile 485)
UI_TradesPanel_OnChartEvent	trades_panel.mqh (Zeile 721)
UI_CloseOnePositionAndNotify	event.mqh (Zeile 456)
TP_FinalizeLineMove	event.mqh (Zeile 25)
UI_CreateOrUpdateLineTag	Trade Assistent V2.008.mq5 (Zeile 1265)

6. Wichtigste Lücken und konkrete Empfehlungen

A) Linien-Drag → UI synchronisieren (Anforderungen 3, 17)

- Fund: PR_HL/SL_HL werden bei Change nur gespeichert (event.mqh, Block um Zeile 175).
- Empfehlung: In OnChartEvent bei PR_HL/SL_HL (Drag/Change) zusätzlich die zugehörigen Buttons/Edits per *ChartTimePriceToXY* neu positionieren und Texte neu setzen (analog movingState_R3/R5).

B) Basis-UI aus DB restaurieren (Anforderung 16, teilweise 10)

- DB_SaveLinePrices() schreibt meta keys *price_entry* / *price_sl* (gui_elemente.mqh, Zeile 101).
- Empfehlung: In **createEntryAndSLLinien()** beim Start erst DB_GetMetaText/DB_GetMetaInt lesen und PR_HL/SL_HL + Entry/SL Buttons + Sabio-Edits mit diesen Werten/Koordinaten initialisieren (statt Chart-Mitte + iClose). Optional zusätzlich X/Y der UI-Gruppe in meta speichern (z. B. *ui_x*, *ui_y*).

C) POSNB wirklich als pos_no akzeptieren (Anforderung 11)

- Fund: POSNB wird zwar angezeigt, pos_no wird aber aus DB_GetNextPosNo-Logik bestimmt und POSNB nach Send überschrieben (discord_send.mqh).
- Empfehlung: POSNB als optionalen Override akzeptieren (nur wenn valid int > 0). Kollisionscheck gegen existierende pos_no in DB (z. B. „pos_no bereits vorhanden → Warnung/Block“).

D) Sabio-Edits persistent machen (Anforderung 12)

- Fund: UpdateSabioTP() speichert nicht in DB und konvertiert Strings nicht korrekt (Trade Assistent V2.008.mq5, Zeile 566).
- Empfehlung: SabioEntry/SabioSL bei ENDEDIT als meta keys speichern (z. B. *sabio_entry/sabio_sl*) und beim Init setzen. Wenn numerisch benötigt: *StringToDouble()* verwenden.

E) Trades-Panel verschiebbar (Anforderung 14)

- Empfehlung: TP_BG als dragbares Objekt markieren und Drag-Delta in OnChartEvent behandeln; danach UI_TradesPanel_RebuildRows() mit neuen TP_X/TP_Y. TP_X/TP_Y in meta persistieren.

F) Export über alle DB-Files (Anforderung 15)

- Empfehlung: Funktion „ExportAllTradesToCSV()“: im Files-Ordner nach *DowHowState_*.sqlite* suchen (FileFindFirst/Next), DB öffnen, positions auslesen, CSV schreiben (für Excel).

G) 'Send & Trade' Broker-Order (Anforderung 18)

- Fund: Broker-Trade-Code ist derzeit auskommentiert.
- Empfehlung: Code reaktivieren/neu implementieren und an SendOnlyButton koppeln; DB-Feld *was_sent* und *is_pending* setzen.

H) SL bereits beim Load → bestätigungsbasierte Meldung (Anforderung 20)

- Empfehlung: Nach DB_RestoreAll() einmal prüfen, welche Positionen bereits SL-bedingt „eigentlich zu“ wären. Statt sofort Discord zu senden: Alert + Panel-Button „SL-Alerts senden“ pro Position/Trade.

7. Anwender-Checkliste (zum Abhaken)

Diese Checkliste ist für manuelle Tests im Terminal gedacht (je Symbol/Timeframe).

- [] 1. EA-Start: rechts 2 Linien (Entry/SL), 2 Buttons (Entry/SL), Send-Button, Sabio-Edits, Trend/Pos Eingaben
- [] 2. Entry/SL-Button verschiebbar: andere Objekte wandern mit; Text zeigt Preis & Lot; Buy↔Sell wenn SL>Entry
- [] 3. Entry/SL-Linien verschiebbar und lösen dieselbe Synchronisierung aus wie Buttons
- [] 4. Links: Panel mit LONG/SHORT Label
- [] 5. Active-Trade Buttons LONG/SHORT unsichtbar; sobald Trade existiert sichtbar (rot)
- [] 6. Cancel-Trade Buttons LONG/SHORT unsichtbar bis Trade existiert
- [] 7. Pro Position: Tabellenzeile mit Infos + Cancel-Button + 'SL erreicht'-Button; zusätzlich Send-Button erzeugt Trade und sendet Discord
- [] 8. Cancel/Stop-Buttons: löschen Trade/Position, senden Discord; pro Trade Entry/SL-Linien mit Label, verschiebbar; Verschieben sendet Discord
- [] 9. Infos der Linien/Trades/Positionen pro Symbol werden in DB gespeichert
- [] 10. Bei EA-Neuladen: aus DB lesen und Zustand wiederherstellen (Linien, Buttons, Panel, etc.)
- [] 11. Anwender kann Trade-Nummer und Pos-Nummer überschreiben; EA nutzt diese fortlaufend
- [] 12. Sabio-Preise: Eingabe in Editfeldern; Werte werden gespeichert und an Discord mitgesendet
- [] 13. Buttons im Panel lösen passende Aktion aus (Grafikobjekte löschen, Trade/Pos in DB schließen, Discord senden)
- [] 14. Trades-Panel verschiebbar; Objekte wandern mit
- [] 15. Tabellenliste aller Trades aus allen DB-Files (Excel-Export) – noch zu bauen
- [] 16. Beim Neuladen: Entry/SL Buttons/Linien + Sabio-Edits sollen aus DB restauriert werden (nicht in Chart-Mitte erscheinen)
- [] 17. Bei Verschieben von SL/Entry Button oder Linie muss sichtbarer Preis korrekt aktualisiert sein
- [] 18. Wenn 'Send & Trade' aktiv: echte Broker-Position senden und in DB markieren
- [] 19. SL erreicht: Discord-Meldung automatisch senden
- [] 20. Nach Load: offene Positionen prüfen; wenn SL erreicht -> Alert + Möglichkeit, dies noch an Discord zu senden