

TradeAssistant V2.012

QA Test Checkliste

(Code-Analyse: 95% erfüllt)

Version: V2.012

Test-Datum: _____

Tester: _____

Testumgebung: MetaTrader 5 Build: _____

Symbol: _____ Timeframe: _____

Code-Status: 19/20 Anforderungen implementiert (95%)

Testanleitung

Diese Checkliste testet die 20 funktionalen Anforderungen des TradeAssistant EA. Die Code-Analyse zeigt, dass 95% der Anforderungen vollständig implementiert sind.

Bewertungskriterien:

- ✓ **PASS** - Funktioniert wie erwartet, keine Probleme
- **TEILWEISE** - Grundfunktion vorhanden, kleinere Abweichungen
- ✗ **FAIL** - Funktioniert nicht oder schwerwiegende Fehler
- N/A **NICHT ANWENDBAR** - Test konnte nicht durchgeführt werden

Code-Status pro Anforderung:

- Anforderungen 1-14, 16-20: **Code vollständig implementiert** ✓
- Anforderung 15: **Teilweise implementiert (Export fehlt)** ■

Wichtig: Alle Tests sollten PASS ergeben, da der Code die Funktionalität implementiert hat. Bei FAIL bitte detaillierte Beschreibung in Bemerkungen notieren.

Test 1: EA Initialisierung und UI-Elemente

Code-Status: ✓ Code vollständig

Vorbedingung:	EA auf Chart laden (Neustart)
Testschritte:	1. EA auf Chart ziehen und aktivieren 2. Warten bis EA vollständig geladen 3. Rechte Seite des Charts prüfen
Erwartetes Ergebnis:	✓ 2 horizontale Linien sichtbar (PR_HL und SL_HL) ✓ EntryButton sichtbar ✓ SLButton sichtbar ✓ SendButton sichtbar
Code-Belege:	OnInit(), UI-Definitionen Zeile 97-119
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 2: Entry und SL Button verschieben

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft, UI-Elemente sichtbar
Testschritte:	1. EntryButton mit Maus anklicken und verschieben 2. Text im Button prüfen (Preis und Lot) 3. SLButton oberer Bereich anklicken
Erwartetes Ergebnis:	✓ EntryButton lässt sich verschieben ✓ Text zeigt aktuellen Preis ✓ Text zeigt berechnetes Lot ✓ Andere Werte sind korrekt
Code-Belege:	event.mqh: DRAG Handler, update_Text() Funktion
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 3: Linien verschieben

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft, Linien sichtbar
Testschritte:	1. Entry-Linie (PR_HL) anklicken 2. Linie nach oben/unten ziehen 3. Prüfen: Buttons/Objekte aktualisieren sich
Erwartetes Ergebnis:	✓ Entry-Linie (PR_HL) lässt sich verschieben ✓ SL-Linie (SL_HL) lässt sich verschieben ✓ Buttons aktualisieren sich
Code-Belege:	event.mqh: Linien-DRAG, OBJ_HLINE
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 4: Trades Panel mit Long/Short Labels

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft
Testschritte:	1. Linke Seite des Charts prüfen 2. Panel identifizieren 3. Labels prüfen
Erwartetes Ergebnis:	✓ Panel sichtbar auf linker Seite ✓ 'LONG' Label vorhanden (TA_OVERVIEW_TXT_LONG) ✓ 'SHORT' Label vorhanden (TA_OVERVIEW_TXT_SHORT)
Code-Belege:	trades_panel.mqh: UI_TradesPanel_Create()
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 5: Active Trade Buttons - Sichtbarkeit und Farbe

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft, kein aktiver Trade
Testschritte:	1. Panel prüfen - Active Trade Buttons Status 2. Trade erstellen (SendButton) 3. Panel nochmal prüfen 4.
Erwartetes Ergebnis:	✓ Vor Trade: Buttons unsichtbar (OBJPROP_TIMEFRAMES) ✓ Nach Trade: Buttons sichtbar ✓ Buttons in RO
Code-Belege:	active_long/short_trade_no, OBJPROP_COLOR
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 6: Cancel Buttons Sichtbarkeit

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft
Testschritte:	1. Panel ohne Trade prüfen 2. Trade erstellen 3. Cancel Buttons identifizieren
Erwartetes Ergebnis:	✓ Ohne Trade: Cancel Buttons unsichtbar ✓ Mit Trade: Cancel Buttons für Long/Short sichtbar ✓ Buttons: TP
Code-Belege:	trades_panel.mqh: Cancel Button Logik
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 7: Position-Zeilen mit Infos und Buttons

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft, Trade mit Positionen
Testschritte:	1. Trade mit 2-3 Positionen erstellen 2. Panel prüfen 3. Zeilen-Inhalte prüfen
Erwartetes Ergebnis:	✓ Pro Position eine Zeile ✓ Trade-Nummer angezeigt ✓ Position-Nummer angezeigt ✓ Entry-Preis angeze
Code-Belege:	UI_TradesPanel_RebuildRows()
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 8: Cancel/Stop Button - Löschen + Discord

Code-Status: ✓ Code vollständig

Vorbedingung:	Trade vorhanden, Discord konfiguriert
Testschritte:	1. Position im Panel identifizieren 2. Cancel-Button (C) klicken 3. Discord prüfen 4. Neue Position erstellen
Erwartetes Ergebnis:	✓ Cancel: Position wird gelöscht ✓ Cancel: Discord-Nachricht gesendet ✓ Stop: Position gestoppt/geschlossen
Code-Belege:	event.mqh: Click Handler, discord_send.mqh
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 9: SendButton - Trade erstellen + Discord

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft, Discord aktiv
Testschritte:	1. Entry/SL setzen 2. Sabio-Preise eingeben 3. Trend-Nr und Position eingeben 4. SendButton klicken
Erwartetes Ergebnis:	✓ Trade wird erstellt ✓ Discord-Nachricht gesendet ✓ Nachricht enthält: Symbol, Entry, SL, Lot ✓ Nachricht enthält: Trend-Nr, Position, Stop-Loss, Take-Profit
Code-Belege:	SENDTRADEBTN Handler, CDiscordClient
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 10: Trade Linien mit Labels - verschiebbar

Code-Status: ✓ Code vollständig

Vorbedingung:	Trade erstellt
Testschritte:	1. Trade erstellen 2. Chart prüfen auf Linien 3. Labels rechts prüfen 4. Linie verschieben 5. Label-Position verschieben
Erwartetes Ergebnis:	✓ Entry_Long_X_Y Linie gezeichnet ✓ SL_Long_X_Y Linie gezeichnet ✓ Labels mit Text 'T# P# ENTRY/SL F#
Code-Belege:	Entry_Long/Short, SL_Long/Short, LINE_TAG_SUFFIX
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 11: Datenbank-Persistierung

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft
Testschritte:	1. Trade erstellen 2. Linien verschieben 3. EA vom Chart entfernen 4. EA neu laden 5. Daten prüfen
Erwartetes Ergebnis:	✓ Trade in SQLite DB gespeichert ✓ Linien-Positionen gespeichert ✓ Nach Neuladen: Trades restauriert
Code-Belege:	CDBService, RestoreFromDB()
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 12: Sabio-Preise - Eingabe und Discord

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft
Testschritte:	1. SabioEntry ausfüllen (z.B. 1.0850) 2. SabioSL ausfüllen (z.B. 1.0800) 3. Trade erstellen 4. Discord-Nachricht senden
Erwartetes Ergebnis:	✓ Sabio-Preise eingabefähig ✓ Werte werden gespeichert ✓ Discord enthält Sabio-Preise ✓ ObjectGetS...
Code-Belege:	SabioEntry/SabioSL, Sabio in Discord
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 13: Panel Buttons - Aktionen ausführen

Code-Status: ✓ Code vollständig

Vorbedingung:	Trade vorhanden
Testschritte:	1. Button im Panel klicken 2. Grafische Objekte prüfen 3. DB prüfen 4. Discord prüfen
Erwartetes Ergebnis:	✓ Click-Event wird verarbeitet ✓ Grafische Objekte gelöscht (ObjectDelete) ✓ DB aktualisiert (UpdateStatus/D
Code-Belege:	CHARTEVENT_OBJECT_CLICK, ObjectDelete
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 14: Panel verschieben

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft
Testschritte:	1. Panel mit Maus anklicken 2. Panel ziehen 3. Objekte prüfen ob sie mitbewegen
Erwartetes Ergebnis:	✓ Panel lässt sich verschieben ✓ Objekte wandern mit ✓ movingState Variablen aktiv ✓ CHARTEVENT_
Code-Belege:	movingState, mlbDown, MOUSE_MOVE Handler
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 15: Tabellenexport aller Trades

Code-Status: ■ Teilweise (Export-Funktion fehlt)

Vorbedingung:	Mehrere Trades in DB
Testschritte:	1. Export-Funktion im EA suchen 2. Falls vorhanden: Export ausführen 3. CSV/Excel-Datei prüfen
Erwartetes Ergebnis:	■ Export-Funktion möglicherweise nicht vorhanden ✓ DB-Infrastruktur für Export ist vorhanden Info: CDBServ
Code-Belege:	DB-Infrastruktur vorhanden, explizite Export-Funktion nicht implementiert
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 16: Neuladen - Button-Position Mitte

Code-Status: ✓ Code vollständig

Vorbedingung:	EA mit Trade
Testschritte:	1. Trade-Linien vorhanden 2. EA entfernen 3. EA neu laden 4. Entry/SL-Button Position prüfen 5. Tra
Erwartetes Ergebnis:	✓ Entry/SL-Buttons in Chart-Mitte ✓ Buttons NICHT aus DB restauriert ✓ Trade-Linien AUS DB restauriert
Code-Belege:	OnInit(), RestoreFromDB(), getChartWidthInPixels()
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 17: Preis im sichtbaren Bereich korrekt

Code-Status: ✓ Code vollständig

Vorbedingung:	EA läuft
Testschritte:	1. Entry-Button verschieben (auch außerhalb) 2. Preis im Button prüfen 3. Chart zoomen/verschieben 4. ...
Erwartetes Ergebnis:	✓ Preis entspricht Button-Position ✓ ChartTimePriceToXY() berechnet korrekt ✓ Preis korrekt formatiert (_Dig
Code-Belege:	ChartTimePriceToXY(), OBJPROP_PRICE
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 18: Echte Broker-Position (Send & Trade)

Code-Status: ✓ Code vollständig

Vorbedingung:	Demokonto, Send & Trade aktiv
Testschritte:	1. 'Send & Trade' Modus aktivieren 2. Trade erstellen 3. MT5 'Trade' Tab prüfen 4. DB prüfen auf Broker-
Erwartetes Ergebnis:	✓ Echte Position am Broker erstellt ✓ CTrade.Buy()/Sell() ausgeführt ✓ Position in MT5 sichtbar ✓ Broke
Code-Belege:	CTrade trade, OrderSend, PositionSelect
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 19: SL-Hit - Discord Benachrichtigung

Code-Status: ✓ Code vollständig

Vorbedingung:	Trade mit SL aktiv
Testschritte:	1. Trade mit SL nahe Preis erstellen 2. Warten bis Preis SL erreicht 3. Discord prüfen 4. Nachricht prüfen
Erwartetes Ergebnis:	✓ Discord-Nachricht bei SL-Hit ✓ Nachricht enthält: Symbol, Trade#, Pos# ✓ Nachricht enthält: BUY oder SELL
Code-Belege:	OnTick(), SLHIT_PREFIX, FormatSLHitMessage
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test 20: Start mit Positionen - SL-Check + Alert

Code-Status: ✓ Code vollständig

Vorbedingung:	Offene Broker-Position vorhanden
Testschritte:	1. Position manuell öffnen 2. SL setzen (nahe Preis) 3. EA laden 4. Alert prüfen 5. Discord-Button prüfen
Erwartetes Ergebnis:	✓ OnInit() prüft existierende Positionen ✓ Falls SL erreicht: Alert angezeigt ✓ MessageBox mit Details ✓
Code-Belege:	OnInit(), PositionsTotal(), Alert(), ButtonSLHit
Testergebnis:	■ PASS ■ TEILWEISE ■ FAIL ■ N/A
Bemerkungen:	

Test-Zusammenfassung

Status	Anzahl	Test-Nummern
PASS		
TEILWEISE		
FAIL		
N/A		
Gesamt	20	Alle

Erwartete Test-Ergebnisse

Basierend auf der Code-Analyse sollten folgende Ergebnisse erzielt werden:

- **Tests 1-14:** Alle sollten PASS sein (Code vollständig implementiert)
- **Test 15:** Kann TEILWEISE sein (Export-Funktion nicht explizit vorhanden)
- **Tests 16-20:** Alle sollten PASS sein (Code vollständig implementiert)

Bei FAIL-Ergebnissen:

Falls ein Test FAIL ergibt, obwohl der Code vollständig ist, könnte es sein:

- Konfigurationsproblem (z.B. Discord Webhook nicht gesetzt)
- Testumgebung-Problem (z.B. Broker-Verbindung)
- Runtime-Bug (Code ist da, aber funktioniert nicht wie erwartet)

Bitte dokumentieren Sie FAIL-Fälle sehr genau mit:

- Exakten Schritten zur Reproduktion
- Fehlermeldungen (falls vorhanden)
- MT5 Terminal Log
- Screenshots

Unterschrift Tester

Datum