ALGOTS
Obfuscator

TECHNISCHE DOKUMENTATION

Benutzerhandbuch

ALGOTS Obfuscator | Deutsch
Windows 10 / 11 (64-bit) • MQL5 / MetaEditor

VERSION 4.2.1 | MQL5 | Windows x64

© ALGOTS

ALGOTS / DEUTSCH

Inhaltsverzeichnis

01 Systemanforderungen ... 3
02 Installation ... 3
03 Lizenz aktivieren ... 4
04 Schnellstart: erster geschützter Build ... 5
05 Desktop-App im Detail ... 5
06 Presets Safe, Harden und Custom ... 6
07 Kompilierprüfung mit MetaEditor ... 7
08 Seeds und kundenspezifische Builds ... 7
09 Reversible Builds und Restore ... 8
10 Inhalt des Ausgabeordners ... 8
11 Kommandozeile (CLI) ... 9
12 Lizenzverwaltung ... 11
13 Sprachen ... 12
14 Dateien und Ordner auf Ihrem PC ... 12
15 Bewährte Vorgehensweisen ... 12
16 Fehlerbehebung ... 13
17 Deinstallation ... 14
18 Support ... 14

> ALGOTS Obfuscator schützt MQL5-Quellcode. Das Programm verarbeitet eine `.mq5`-Projektdatei samt eingebundenen `.mqh`-Dateien und erzeugt Quellcode, der weiterhin in MetaEditor kompilierbar sein und das Programmverhalten beibehalten soll, jedoch deutlich schwerer lesbar, kopierbar und wiederverwendbar ist. Kompilierung und Funktionstests bleiben erforderlich. Lesbare Bezeichner werden ersetzt, Kommentare und Formatierungen können entfernt werden; stärkere Modi schützen zusätzlich Zeichenfolgen, Konstanten und Kontrollfluss.
>
> Die Verarbeitung erfolgt vollständig lokal auf Ihrem PC. Ihr Quellcode wird niemals hochgeladen.

1. Systemanforderungen

Anforderung	Beschreibung
Betriebssystem	Windows 10 oder Windows 11, 64 Bit
Berechtigungen	Normales Benutzerkonto; keine Administratorrechte erforderlich
Internet	Für Aktivierung und regelmäßige Lizenzprüfungen erforderlich
MetaTrader 5	Optional, aber empfohlen: Der 64-Bit-MetaEditor (`metaeditor64.exe`) führt die automatische Kompilierprüfung aus
Speicherplatz	Etwa 100 MB

Python oder eine andere Laufzeitumgebung ist nicht erforderlich; alle Komponenten sind enthalten.

2. Installation

1. Starten Sie `ALGOTS-Obfuscator-4.2.1-Setup.exe`.
2. Wählen Sie die Sprache des Installationsprogramms aus den 13 unterstützten Sprachen.
3. Auf **Select Additional Tasks** können Sie folgende Aufgaben auswählen:
   - **Create a desktop shortcut**: optionaler Desktop-Link.
   - **Add the CLI to the user PATH (command: algots-obfuscator)**: `algots-obfuscator` wird in neuen PowerShell- oder Eingabeaufforderungsfenstern ausführbar. Nur der Benutzer-PATH wird geändert; Administratorrechte sind nicht nötig.
4. Schließen Sie den Assistenten ab.

Standardmäßig wird die Anwendung nur für den aktuellen Benutzer unter `%LOCALAPPDATA%\Programs\ALGOTS Obfuscator` installiert. Im Startmenü finden Sie **ALGOTS Obfuscator** (Desktop-App), **ALGOTS Obfuscator CLI** (Terminal mit Befehlsübersicht) und **Uninstall ALGOTS Obfuscator**.

Update: Führen Sie eine neuere Installationsdatei aus. Die Programmdateien werden ersetzt, Einstellungen, Lizenz und private Build-Daten bleiben erhalten.

> **Windows SmartScreen.** Erscheint „Windows protected your PC“, prüfen Sie Downloadquelle, gegebenenfalls Angaben zu Herausgeber und Signatur sowie die Dateiintegrität. Laden Sie ausschließlich von algots.net oder über den offiziellen Kauflink herunter. Können Sie den Installer nicht verifizieren oder bleibt die Warnung bestehen, kontaktieren Sie support@algots.net vor dem Start. Umgehen Sie keine Windows-Schutzmaßnahmen.

> **Paketintegrität.** Bei jedem Start prüft die Anwendung ihre eigenen Dateien. „Package integrity check failed. Reinstall the complete application.“ kann auf eine veränderte oder entfernte Datei hinweisen, unter anderem durch Antivirus-Quarantäne. Installieren Sie das Paket neu; bei einer Quarantäne kontaktieren Sie den Support zur Prüfung von Herkunft und Integrität. Deaktivieren Sie keine Schutzfunktionen.

3. Lizenz aktivieren

Ihren Lizenzschlüssel erhalten Sie nach dem Kauf per E-Mail; er steht auch auf der Lemon-Squeezy-Bestellseite.

Desktop-App: Öffnen Sie **ALGOTS Obfuscator**, klicken Sie oben auf **License**, fügen Sie den Schlüssel in **License key** ein und klicken Sie auf **Activate License**. Anschließend werden **OBFUSCATE** und **RESTORE** verfügbar.

Kommandozeile:

```
algots-obfuscator license activate
```

Die Schlüsselabfrage ist verdeckt; er wird nicht als Befehlsargument übergeben und erscheint daher nicht im Shell-Verlauf.

An den Lizenzdienst übertragene Daten: Nur Lizenzschlüssel sowie Gerätekennung/-name werden per HTTPS an Lemon Squeezy gesendet. Quellcode, Projektpfade, Seeds und Build-Ergebnisse verbleiben auf Ihrem PC. Die App sendet keine Telemetrie.

Eine Lizenz darf gleichzeitig auf einer begrenzten Zahl von Geräten aktiv sein. Vor dem Wechsel auf einen neuen PC deaktivieren Sie sie auf dem alten (Abschnitt 12).

4. Schnellstart: erster geschützter Build

1. **Hauptdatei:** Klicken Sie neben **Main EA (.mq5)** auf **Browse...** und wählen Sie die Haupt-`.mq5`-Datei Ihres Expert Advisors, Indikators oder Skripts. Zugehörige `#include`-Dateien werden automatisch gesucht und analysiert.
2. **Ausgabeordner:** Wählen Sie einen **leeren** Ordner, niemals Ihren Quellordner.
3. **Preset:** Verwenden Sie für den ersten Build **Safe**.
4. **Kompilierprüfung:** Lassen Sie **Compile automatically with MetaEditor after obfuscation** aktiviert. Klicken Sie bei **MetaEditor** und **MQL5 root** auf **Auto**, falls die Pfade nicht erkannt wurden.
5. Klicken Sie auf **OBFUSCATE**.

Nach Abschluss zeigt die Statusleiste das Kompilierergebnis; **Open output** öffnet die geschützten Dateien. Testen Sie das geschützte Programm vor jeder Weitergabe in MetaTrader 5, z. B. im Strategy Tester, genauso gründlich wie das Original.

5. Desktop-App im Detail

Feld / Schaltfläche	Funktion
Main EA (.mq5)	Haupt-`.mq5`-Datei von EA, Indikator oder Skript; unveränderte Bezeichnung der englischen Oberfläche
Output folder	Ordner für das geschützte Projekt
MetaEditor	Pfad zu `metaeditor64.exe`; **Auto** versucht ihn zu erkennen
MQL5 root	`MQL5`-Datenordner des Terminals mit `Include\`; nötig für `#include <...>` und Kompilierung; **Auto** versucht die Erkennung
Preset	Safe, Harden oder Custom
Options	Einzelne Schutzmaßnahmen, nur bei Custom änderbar
Build / recovery seed	Optionaler Seed für reproduzierbare Builds oder Wiederherstellung; **Generate** erstellt einen sicheren Zufalls-Seed
Reversible build	Bettet eine verschlüsselte Kopie des Originals ein, die mit dem Seed wiederhergestellt werden kann
Allow absolute `__PATH__`	Siehe Fehlerbehebung; standardmäßig ausgeschaltet
Compile automatically with MetaEditor	Kompiliert den geschützten Quellcode vor dem Speichern

Feld / Schaltfläche	Funktion
OBFUSCATE	Startet einen geschützten Build
RESTORE	Stellt das Original aus einem reversiblen Build wieder her
Open output / Clear log	Öffnet den Ausgabeordner / löscht die Protokollanzeige
License	Aktivierung, Status, Onlineprüfung, Geräte-Deaktivierung
Language / Dark theme	Sprache und Hell-/Dunkelmodus

Pfade, Preset, Optionen, Sprache und Design werden für spätere Sitzungen gespeichert. Seeds und Lizenzschlüssel werden niemals in den Einstellungen gesichert.

6. Presets Safe, Harden und Custom

Der Motor folgt dem Grundsatz: **Wenn eine Transformation für einen Codeabschnitt nicht sicher nachgewiesen werden kann, bleibt dieser unverändert.** Ein weniger aggressiver Build, der Kompilierung und Funktionstests besteht, ist besser als eine Transformation, die Regressionen verursacht.

Safe (Standard, empfohlen)

Benennt eindeutige projektspezifische Symbole um: Funktionen, Parameter, lokale und globale Variablen, Klassen, Strukturen, Enums und Mitglieder. MQL5-Ereignisbehandler (`OnInit`, `OnTick`, `OnCalculate`, …), `export`-Funktionen, eingebaute Namen und mehrdeutige Konstrukte bleiben erhalten. Kommentare und Formatierung bleiben ebenfalls erhalten. **`input`-Namen werden nicht geändert**, damit EA-Eigenschaftsdialog und bestehende `.set`-Dateien weiterhin funktionieren.

Verwenden Sie Safe im Alltag und immer als Erstversuch bei einem neuen Projekt.

Harden

Enthält Safe und zusätzlich:

- **Flatten includes:** Fasst das Projekt in einer geschützten Quelldatei zusammen.
- **Strip comments** und **Minify:** Entfernen Kommentare und Formatierung.
- **Protect strings:** Verhindert lesbar gespeicherte Zeichenfolgenliterale.
- **Encode constants:** Kodiert ganzzahlige Konstanten.
- **Control-flow hardening:** Ergänzt undurchsichtige Bedingungen zur Verschleierung des Ablaufs.

Auch Harden belässt `input`-Namen unverändert. Nutzen Sie es für Auslieferungen, nachdem Safe korrekt kompiliert wurde und die Funktionstests bestanden hat.

Custom

Hier lassen sich flatten includes, minify, strip comments, protect strings, encode constants, control-flow hardening und rename inputs einzeln aktivieren. Zeichenfolgen-, Konstanten- und Kontrollflussschutz setzen **Flatten includes** voraus.

> **Rename inputs** (nur Custom) benennt auch `input`-Variablen um. Das Programm kann weiterhin laufen, aber die Parameternamen im EA-Dialog ändern sich und ältere `.set`-Dateien passen nicht mehr. Lassen Sie die Option aus, sofern dies nicht ausdrücklich gewünscht ist.

Die CLI bietet im Custom-Modus zusätzliche Kontrolle, darunter Namensausschlüsse über `--protect-name` und die Auswahl umzubenennender Symbolkategorien.

7. Kompilierprüfung mit MetaEditor

Bei aktivierter Prüfung kompiliert ALGOTS Obfuscator den geschützten Quellcode mit MetaEditor **vor** dem Schreiben in den Ausgabeordner. Nur bei null Fehlern und einer neu erzeugten `.ex5`-Datei wird der Build angenommen. Bei einem Fehler bleibt der vorherige Ordnerinhalt unverändert.

- Automatisch erkannt wird ausschließlich die 64-Bit-Version von MetaEditor. Bei mehreren MetaTrader-Installationen wählen Sie diese mit **Browse…** aus.
- **MQL5 root** muss der Datenordner desselben Terminals sein (MetaEditor: **File → Open Data Folder**, anschließend `MQL5`). Schreibberechtigung ist notwendig, da darin ein temporärer Ordner angelegt wird.

Die Prüfung belegt die Kompilierung in dieser Konfiguration, **nicht identisches Laufzeitverhalten in allen Situationen**. Tests in MetaTrader bleiben erforderlich.

8. Seeds und kundenspezifische Builds

Jeder Build verwendet einen Seed für die Erzeugung von Namen und Schutzmaßnahmen.

- **Leeres Seed-Feld:** Es wird ein neuer Zufalls-Seed verwendet, daher unterscheiden sich Builds.
- **Eigener Seed:** Bei gleichem Projekt, Preset und Seed entsteht dieselbe Ausgabe; damit können Sie ausgelieferte Builds reproduzieren.
- **Kundenspezifische Builds:** Verwenden Sie für jeden Kunden einen anderen Seed. Dadurch entstehen unterschiedlich geschützte Kopien, die sich unterscheiden lassen.

Dokumentieren Sie die Zuordnung Kunde–Seed intern und behandeln Sie Seeds vertraulich.

9. Reversible Builds und Restore

Normalerweise sind Builds **nicht reversibel**. Die Ausgabe enthält nur geschützten Code; das Original lässt sich daraus nicht wiederherstellen. Bewahren Sie den Originalquellcode in einem Backup oder Versionsverwaltungssystem auf.

Mit **Reversible build** enthält die Ausgabe zusätzlich eine verschlüsselte und authentifizierte Kopie des Originals (AES-GCM, aus dem Seed abgeleiteter Schlüssel). Hierfür benötigen Sie einen sicheren Zufalls-Seed: **Generate** erzeugt einen 256-Bit-Seed. Der Seed wird ebenfalls im privaten lokalen Vault gespeichert (Abschnitt 14).

Wiederherstellung:

1. Wählen Sie die geschützte Hauptdatei `.mq5`.
2. Geben Sie einen leeren Ausgabeordner für das wiederhergestellte Projekt an.
3. Tragen Sie den zugehörigen Seed ein.
4. Klicken Sie auf **RESTORE**.

Die Wiederherstellung erfolgt bytegenau, einschließlich Kommentare, Formatierung und Dateikodierung. Bei falschem Seed oder veränderter Datei wird der Vorgang vor dem Schreiben abgelehnt.

> **Ohne Seed kann niemand – auch ALGOTS nicht – einen reversiblen Build wiederherstellen.** Sichern Sie ihn sorgfältig und geben Sie ihn niemals an Kunden weiter.

Liefern Sie an Kunden vorzugsweise nicht reversible Builds.

10. Inhalt des Ausgabeordners

- Geschütztes Projekt zum Kompilieren oder Ausliefern. Mit **Flatten includes** kann die Hauptdatei `YourEA_obfuscated.mq5` heißen.
- `build_report.json`: **öffentlicher** Bericht mit Status, Anzahl der transformierten und erhaltenen Symbole, Warnungen und Kompilierergebnis. Er enthält keine Originalnamen, Symbolkarte, absoluten Pfade oder Seeds.

Private Daten (Symbolkarte Originalname ↔ neuer Name, detaillierter Diagnosebericht und Seeds reversibler Builds) werden im lokalen Vault auf dem PC gespeichert, nicht im Ausgabeverzeichnis. Verwenden Sie sie nur intern und geben Sie sie niemals weiter.

Ordnerregeln: Der Ausgabeordner muss leer oder früher von ALGOTS Obfuscator erstellt worden sein. Bei jedem Build wird er vollständig und sicher ersetzt, ohne verwaiste Altdateien. Schlägt ein Build fehl, wird der vorherige Zustand wiederhergestellt. Ordner mit fremden Daten werden gegen versehentliches Überschreiben abgelehnt.

11. Kommandozeile (CLI)

CLI und Desktop-App verwenden denselben Motor und dieselbe Lizenz. Die CLI eignet sich für Skripte, Batch-Builds und CI-Pipelines.

Start: Wenn Sie **Add the CLI to the user PATH** gewählt haben, öffnen Sie ein **neues** PowerShell- oder Eingabeaufforderungsfenster und geben `algots-obfuscator` ein. Andernfalls verwenden Sie **ALGOTS Obfuscator CLI** im Startmenü oder `"%LOCALAPPDATA%\Programs\ALGOTS Obfuscator\CLI\ALGOTSObfuscatorCLI.exe"`.

Befehle

Befehl	Zweck
`obfuscate MAIN -o OUTPUT`	Erstellt eine geschützte Projektkopie
`restore MAIN -o OUTPUT`	Stellt das Original aus einem reversiblen Build wieder her
`inspect MAIN`	Listet Quellen, Includes, Ressourcen und fehlende Abhängigkeiten, ohne Dateien zu schreiben
`doctor [MAIN]`	Prüft Umgebung, MetaEditor, MQL5-Roots und Komponenten
`wizard`	Interaktiver Assistent mit abschließender Bestätigung
`license ...`	`activate`, `status`, `check`, `deactivate`, `forget-local`
`language [CODE]`	Zeigt/ändert die gemeinsame Sprache von CLI und Desktop-App
`open-output PATH`	Öffnet einen Ausgabeordner im Explorer
`version`, `about`	Version und Produktinformationen

Jeder Befehl unterstützt `--help`.

PowerShell-Beispiele:

```
# Guided build
algots-obfuscator wizard

# Safe build with compile check
algots-obfuscator obfuscate "C:\Projects\MyEA\MyEA.mq5" -o "C:\Builds\MyEA" --preset safe --compile

# Harden build with an explicit MetaEditor and MQL5 root
algots-obfuscator obfuscate .\MyEA.mq5 -o .\build-harden --preset harden `
    --metaeditor "C:\Program Files\MetaTrader 5\metaeditor64.exe" `
    --mql5-root "C:\Users\me\AppData\Roaming\MetaQuotes\Terminal\<ID>\MQL5"

# Custom build: keep one name readable
algots-obfuscator obfuscate .\MyEA.mq5 -o .\build-custom --preset custom `
    --flatten-includes --minify --strip-comments --encrypt-strings --protect-name Calculate
```

```
# Repeatable per-customer build
algots-obfuscator obfuscate .\MyEA.mq5 -o .\customer-042 --preset harden --seed "customer-042-2026"

# Reversible build, then restore (the seed is asked with a hidden prompt)
algots-obfuscator obfuscate .\MyEA.mq5 -o .\reversible --recovery --show-recovery-seed
algots-obfuscator restore .\reversible\MyEA.mq5 -o .\restored --ask-seed

# Inspect a project and check the environment
algots-obfuscator inspect .\MyEA.mq5
algots-obfuscator doctor
```

Custom-Optionen, jeweils auch mit `--no-…`: `--strip-comments`, `--flatten-includes`, `--minify`, `--encrypt-strings`, `--encode-constants`, `--opaque-if`, `--rename-parameters`, `--rename-locals`, `--rename-globals`, `--rename-members`, `--rename-inputs`.

In Skripten akzeptiert `restore` auch `--seed-file PATH` (UTF-8-Datei mit Seed). Verwenden Sie `--seed` für Recovery-Seeds möglichst nicht auf gemeinsam genutzten Computern: Befehlszeilen können im Shell-Verlauf oder in Prozesslisten sichtbar sein.

Automatisierung

Option	Funktion
`--json`	Schreibt auch bei Fehlern genau ein JSON-Objekt auf stdout
`--quiet` / `-q`	Unterdrückt Zusammenfassungen; Warnungen und Fehler bleiben auf stderr
`--verbose` / `-v`	Zeigt Motorprotokolle und Diagnosen
`--color auto|always|never`	Terminalfarben
`--lang CODE`	Sprache nur für diesen Befehl, z. B. `--lang en`

Jedes JSON-Ergebnis enthält `schema_version`, `ok`, `command`, `exit_code` und `result` oder `error`. Warnungen stehen in `result.warnings`.

Exit-Codes

Code	Bedeutung
0	Abgeschlossen; Warnungen möglich
1	Unerwarteter interner Fehler
2	Ungültige Befehlszeilenargumente
3	Ungültige Eingabe, Konfiguration, Dateisystem oder Ausgabeordner
4	Ungültiger Toolchain-/Umgebungspfad oder mehrdeutige MQL5-Root

Code	Bedeutung
5	MetaEditor-Kompilierprüfung fehlgeschlagen oder Zeitlimit überschritten
6	Restore verweigert: falscher Seed, veränderte Datei oder ungültiges Ziel
10	Lizenzaktivierung erforderlich
11	Online-Lizenzprüfung erforderlich
20	Lizenz abgelehnt: abgelaufen, deaktiviert oder nicht für das Produkt gültig
30	Lizenzdienstfehler; später erneut versuchen
40	Kein Zugriff auf lokale Lizenzdaten
50	Produktkonfigurationsproblem; Support kontaktieren
51	Paketintegritätsprüfung fehlgeschlagen; neu installieren
130	Unterbrochen (Ctrl+C)

12. Lizenzverwaltung

Klicken Sie in der Desktop-App auf **License** bzw. verwenden Sie in der CLI `license`.

Aktion	Desktop-App	CLI
Aktivieren	**Activate License**	`license activate`
Status anzeigen	Im Dialog sichtbar	`license status`
Onlineprüfung erzwingen	**Check online**	`license check`
Gerät freigeben	**Deactivate This Device**	`license deactivate`
Nur lokale Daten löschen	—	`license forget-local --yes`

- **PC wechseln:** Lizenz zuerst auf dem alten Gerät deaktivieren, dann auf dem neuen aktivieren.
- `forget-local` löscht nur lokale Lizenzdaten und gibt die Serveraktivierung **nicht** frei; dazu `deactivate` verwenden.
- Die Lizenz wird regelmäßig online überprüft. Ist der Dienst nicht erreichbar, kann die Arbeit noch begrenzte Zeit seit der letzten erfolgreichen Prüfung fortgesetzt werden (im Dialog **Offline use allowed until**). Danach Internet verbinden und **Check online** wählen.
- Das Zurückstellen der Systemuhr verlängert die Offlinenutzung nicht.

In automatisierten Umgebungen liest `license activate --key-stdin` den Schlüssel aus stdin. Die Umgebungsvariable `MQL5OBF_LICENSE_KEY` wird ebenfalls unterstützt.

13. Sprachen

Desktop-App, CLI und Installer stehen in Englisch, Italienisch, Spanisch, Französisch, Deutsch, Portugiesisch, Türkisch, Russisch, vereinfachtem Chinesisch, Japanisch, Koreanisch, Arabisch und Persisch bereit.

Wählen Sie die Sprache mit **Language**. Die CLI übernimmt diese Auswahl; andernfalls wird die Windows-Sprache und bei Bedarf Englisch verwendet.

```
algots-obfuscator language        # list languages
algots-obfuscator language it     # set Italian (also for the desktop app)
algots-obfuscator language auto   # follow Windows again
```

Befehlsnamen, Optionen, Exit-Codes und JSON-Schlüssel werden niemals übersetzt.

14. Dateien und Ordner auf Ihrem PC

Speicherort	Inhalt
`%LOCALAPPDATA%\Programs\ALGOTS Obfuscator`	Anwendung
`%APPDATA%\MQL5Obfuscator\settings.json`	Einstellungen
`%LOCALAPPDATA%\MQL5Obfuscator\Licensing`	Lizenzstatus, für Ihr Konto mit Windows DPAPI verschlüsselt
`%LOCALAPPDATA%\MQL5Obfuscator\Builds`	Private Symbolkarten und Diagnoseberichte
`%LOCALAPPDATA%\MQL5Obfuscator\Recovery`	Seeds und Verlauf reversibler Builds

Die Ordner `Builds` und `Recovery` sind privat. Sichern Sie sie, wenn Sie sie für Support oder Restore benötigen, und übermitteln Sie sie niemals an Kunden.

15. Bewährte Vorgehensweisen

- Originalquellen in Versionsverwaltung oder Backup aufbewahren: ALGOTS Obfuscator ist kein Sicherungswerkzeug.
- Jedes neue Projekt zunächst mit **Safe**, danach mit **Harden** verarbeiten.

- Kompilierprüfung aktiviert lassen und das geschützte Programm vor Auslieferung in MetaTrader testen.
- Kunden nur nicht reversible Builds liefern.
- Unterschiedliche Seeds je Kunde verwenden und privat zuordnen.
- Vault, Symbolkarten und Seeds niemals weitergeben.
- Obfuskation erschwert Lesen und Wiederverwenden, verhindert Reverse Engineering aber nicht vollständig. Nutzen Sie für wertvollen Code zusätzlich geeignete Lizenzierung und Verträge.

16. Fehlerbehebung

OBFUSCATE und RESTORE deaktiviert: Lizenz inaktiv oder Onlineprüfung erforderlich. Unter **License** die Option **Activate License** oder **Check online** wählen.

„MetaEditor was not found" / „select the terminal MQL5 root": Mit **Auto** suchen oder `metaeditor64.exe` und den `MQL5`-Ordner manuell auswählen. Die automatische Kompilierprüfung kann auch deaktiviert werden.

Build mit Kompilierfehlern: Zuerst prüfen, ob das Original im selben MetaEditor fehlerfrei kompiliert. Dann Safe testen. Funktioniert Safe, Harden aber nicht, nur den öffentlichen Bericht `build_report.json` an den Support senden, niemals den privaten Vault.

`#include <...>`-Dateien fehlen: MQL5-Root des Terminals auswählen, in dem sie liegen.

Ausgabeordner abgelehnt: Leeren Ordner oder einen von ALGOTS Obfuscator erstellten Ordner verwenden.

Abbruch wegen `__PATH__`: Dieser Wert würde den absoluten Quellpfad (möglicherweise einschließlich Benutzernamen) in ausgelieferten Code schreiben. Entfernen oder **Allow absolute `__PATH__`** nur bei bewusster Zustimmung aktivieren.

„Package integrity check failed": Neu installieren. Wurden Dateien vom Antivirus in Quarantäne gestellt, kontaktieren Sie support@algots.net für Diagnose und Prüfung von Herkunft/Integrität. Schutzmaßnahmen nicht umgehen.

`algots-obfuscator` unbekannt: Neues Terminal öffnen. Falls nötig, Installation mit **Add the CLI to the user PATH** wiederholen.

„activation limit reached": Lizenz auf einem ungenutzten Gerät deaktivieren oder Support kontaktieren.

17. Deinstallation

Unter **Settings → Apps → Installed apps → ALGOTS Obfuscator → Uninstall** oder über **Uninstall ALGOTS Obfuscator** im Startmenü deinstallieren. Der CLI-Pfad wird ebenfalls aus dem Benutzer-PATH entfernt.

Für einen PC-Wechsel die Lizenz **vor der Deinstallation deaktivieren**. Einstellungen und Vault bleiben bestehen; bei Bedarf entfernen Sie die in Abschnitt 14 genannten `MQL5Obfuscator`-Ordner gesondert.

18. Support

- E-Mail: support@algots.net
- Telegram: @ALGOTS_Support
- Web: https://algots.net/support

Geben Sie die Version (`algots-obfuscator version`), Ihre Windows-Version und bei Build-Problemen den öffentlichen `build_report.json` an. Übermitteln Sie niemals Seeds, Symbolkarten oder Inhalte des privaten Vaults.

MetaTrader und MQL5 sind Marken von MetaQuotes Ltd. ALGOTS Obfuscator ist ein unabhängiges Produkt und steht in keiner Verbindung zu MetaQuotes.