ALGOTS
Obfuscator

DOCUMENTAZIONE TECNICA

Guida utente

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

VERSION 4.2.1 | MQL5 | Windows x64

© ALGOTS

ALGOTS / ITALIANO

Indice

01 Requisiti 3

02 Installazione 3

03 Attivazione della licenza 4

04 Primo avvio: creare una build protetta 5

05 L'applicazione desktop nel dettaglio 5

06 Preset: Safe, Harden e Custom 6

07 Controllo di compilazione con MetaEditor 7

08 Seed e build per singolo cliente 7

09 Build reversibili e Restore 8

10 Contenuto della cartella di output 8

11 Interfaccia a riga di comando (CLI) 9

12 Gestione della licenza 11

13 Lingue 12

14 File e cartelle sul PC 12

15 Buone pratiche 13

16 Risoluzione dei problemi 13

17 Disinstallazione 14

18 Assistenza 14

> ALGOTS Obfuscator protegge il codice sorgente MQL5. Elabora un progetto `.mq5` e i relativi file `#include .mqh`, producendo un codice sorgente progettato per continuare a compilare in MetaEditor e a preservare il comportamento del programma, ma molto più difficile da leggere, copiare e riutilizzare. La compilazione e i test funzionali rimangono necessari. Gli identificatori leggibili vengono sostituiti, commenti e formattazione possono essere rimossi e, nelle modalità più avanzate, stringhe, costanti e flusso di controllo ricevono ulteriori protezioni.
>
> L'elaborazione del progetto avviene interamente sul PC. Il codice sorgente non viene caricato su server esterni.

1. Requisiti

Requisito	Dettagli
Sistema operativo	Windows 10 o Windows 11, 64 bit
Permessi	Normale account utente; non sono richiesti privilegi di amministratore
Internet	Necessario per attivare la licenza e per i controlli periodici
MetaTrader 5	Facoltativo ma consigliato: MetaEditor a 64 bit (`metaeditor64.exe`) viene utilizzato per il controllo automatico della compilazione
Spazio su disco	Circa 100 MB

ALGOTS Obfuscator non richiede Python né altri runtime: tutti i componenti necessari sono inclusi.

2. Installazione

1. Esegui `ALGOTS-Obfuscator-4.2.1-Setup.exe`.
2. Scegli la lingua del programma di installazione, disponibile nelle stesse 13 lingue dell'applicazione.
3. Nella schermata **Select Additional Tasks** seleziona, se necessario:
   - **Create a desktop shortcut**: crea un collegamento sul desktop.
   - **Add the CLI to the user PATH (command: algots-obfuscator)**: permette di avviare la CLI digitando `algots-obfuscator` in qualsiasi nuova finestra PowerShell o Prompt dei comandi. Modifica soltanto il PATH dell'utente e non richiede privilegi amministrativi.
4. Completa la procedura guidata.

L'installazione è riservata all'utente corrente e, per impostazione predefinita, viene eseguita in `%LOCALAPPDATA%\Programs\ALGOTS Obfuscator`. Nel menu Start vengono creati **ALGOTS Obfuscator** (app desktop), **ALGOTS Obfuscator CLI** (terminale con la guida ai comandi) e **Uninstall ALGOTS Obfuscator**.

Aggiornamenti: esegui un installer più recente. I file del programma vengono sostituiti mantenendo preferenze, licenza e dati di build privati.

> **Windows SmartScreen.** Se viene mostrato l'avviso «Windows protected your PC», verifica la provenienza del download, i dati dell'editore o della firma (se disponibili) e l'integrità del file. Scarica l'installer soltanto da algots.net o dal collegamento ufficiale di acquisto. Se non riesci a verificarlo o l'avviso persiste, contatta support@algots.net prima di eseguirlo: non aggirare le protezioni di Windows.

> **Integrità del pacchetto.** A ogni avvio l'app verifica che i propri file non siano stati alterati. Se appare «Package integrity check failed. Reinstall the complete application.», un file potrebbe essere stato modificato o rimosso, per esempio dalla quarantena antivirus. Reinstalla utilizzando l'installer; in caso di quarantena contatta support@algots.net per la diagnosi e verifica provenienza e integrità prima di agire. Non disabilitare le protezioni.

3. Attivazione della licenza

Dopo l'acquisto ricevi la chiave di licenza via email; è disponibile anche nella pagina dell'ordine su Lemon Squeezy.

Applicazione desktop

1. Apri **ALGOTS Obfuscator**.
2. Fai clic su **License** nella barra superiore.
3. Incolla la chiave in **License key** e premi **Activate License**.

I pulsanti **OBFUSCATE** e **RESTORE** si attivano quando la licenza è valida.

Riga di comando

```
algots-obfuscator license activate
```

La chiave viene richiesta tramite input nascosto, non come parametro della riga di comando: in questo modo non viene registrata nella cronologia della shell.

Dati inviati al server di licenza. Soltanto la chiave e l'identificativo/nome del dispositivo vengono trasmessi via HTTPS al servizio licenze Lemon Squeezy. Codice sorgente, percorsi dei progetti, seed e risultati delle build restano sul PC. L'applicazione non dispone di telemetria.

Ogni licenza può essere attiva contemporaneamente su un numero limitato di dispositivi. Per trasferirla su un altro PC, disattivala prima sul precedente (vedi la sezione 12).

4. Primo avvio: creare una build protetta

1. **File principale:** accanto a **Main EA (.mq5)** premi **Browse...** e scegli il `.mq5` principale del progetto, che può essere un Expert Advisor, un indicatore o uno script. Devi scegliere soltanto quel file: le dipendenze `#include` vengono individuate e analizzate automaticamente.
2. **Cartella di destinazione:** scegli una cartella **vuota** per la copia protetta. Non utilizzare mai la cartella dei sorgenti.
3. **Preset:** per la prima build mantieni **Safe**.
4. **Verifica della compilazione:** mantieni attivo **Compile automatically with MetaEditor after obfuscation**. Se necessario premi **Auto** accanto a **MetaEditor** e **MQL5 root** per individuarne i percorsi.
5. Premi **OBFUSCATE**.

Al termine, la barra di stato mostra l'esito del controllo di compilazione. Usa **Open output** per aprire i file protetti.

Prima di distribuire il risultato, collauda il programma protetto in MetaTrader 5, ad esempio nello Strategy Tester, con la stessa accuratezza usata per l'originale.

5. L'applicazione desktop nel dettaglio

Campo o pulsante	Funzione
Main EA (.mq5)	File sorgente `.mq5` principale da proteggere; accetta EA, indicatori e script. Il nome indicato è quello attuale dell'interfaccia inglese
Output folder	Cartella in cui viene scritto il progetto protetto
MetaEditor	Percorso di `metaeditor64.exe` per la compilazione; **Auto** ne tenta il rilevamento
MQL5 root	Cartella dati `MQL5` del terminale, contenente `Include\`; necessaria per risolvere `#include <...>` e compilare. **Auto** prova a rilevarla
Preset	Safe, Harden o Custom
Options	Singole trasformazioni, modificabili soltanto con Custom
Build / recovery seed	Seed facoltativo per build riproducibili oppure seed di ripristino; **Generate** crea un valore casuale sicuro

Campo o pulsante	Funzione
Reversible build	Include nell'output una copia cifrata dell'originale, recuperabile con il seed
Allow absolute `__PATH__`	Opzione illustrata nella risoluzione dei problemi; disattivata per impostazione predefinita
Compile automatically with MetaEditor	Compila l'output protetto prima di salvarlo
OBFUSCATE	Avvia la build protetta
RESTORE	Ripristina l'originale da una build reversibile
Open output / Clear log	Apre la cartella di output / cancella il registro
License	Attivazione, stato, verifica online e disattivazione del dispositivo
Language / Dark theme	Lingua dell'interfaccia e tema chiaro/scuro

Percorsi, preset, opzioni, lingua e tema vengono ricordati nella sessione successiva. Seed e chiavi di licenza non vengono mai conservati nelle impostazioni.

6. Preset: Safe, Harden e Custom

ALGOTS Obfuscator segue una regola conservativa: **se non è possibile dimostrare la sicurezza di una trasformazione per una porzione di codice, questa rimane invariata.** Una protezione meno aggressiva che superi compilazione e test funzionali è preferibile a trasformazioni che causano regressioni.

Safe (predefinito, consigliato)

Rinomina, quando l'identificazione è certa, i simboli definiti dal progetto: funzioni, parametri, variabili locali e globali, classi, strutture, enumerazioni e membri. Restano invariati gli event handler MQL5 (`OnInit`, `OnTick`, `OnCalculate`, ...), le funzioni `export`, i nomi integrati e i costrutti ambigui. Commenti e formattazione rimangono. **I nomi delle variabili `input` sono mantenuti**, per preservare la finestra delle proprietà dell'EA e la compatibilità dei file `.set` già esistenti.

Usa Safe per le build quotidiane e come primo preset per ogni nuovo progetto.

Harden

Include tutto ciò che fa Safe e aggiunge:

- **Flatten includes:** unisce il progetto in un unico sorgente protetto.
- **Strip comments** e **Minify:** rimuovono commenti e formattazione.
- **Protect strings:** impedisce di memorizzare le stringhe letterali in forma leggibile.
- **Encode constants:** codifica le costanti intere.

- **Control-flow hardening:** inserisce condizioni opache che rendono più arduo seguire la logica.

Anche Harden mantiene i nomi `input`. Usalo per le copie da distribuire, dopo che il progetto ha superato compilazione e test con Safe.

Custom

Consente di abilitare e disabilitare individualmente flatten includes, minify, strip comments, protect strings, encode constants, control-flow hardening e rename inputs. La protezione di stringhe, costanti e flusso di controllo richiede **Flatten includes**.

> **Rename inputs** (solo Custom) rinomina anche le variabili `input`. Il programma può funzionare, ma cambiano i nomi mostrati nelle proprietà dell'EA e i vecchi file `.set` non corrispondono più. Mantieni l'opzione disattivata salvo che questo sia il risultato desiderato.

La CLI consente controlli ancora più precisi in Custom, come escludere identificatori dalla rinomina con `--protect-name` e scegliere le categorie di simboli da rinominare.

7. Controllo di compilazione con MetaEditor

Se il controllo è attivo, ALGOTS Obfuscator compila il progetto protetto con MetaEditor **prima** di scriverlo nella cartella di output. La build viene accettata soltanto se MetaEditor restituisce zero errori e produce un nuovo file `.ex5`. Se la compilazione fallisce, il contenuto precedente della cartella di output non viene modificato.

- Viene rilevato automaticamente soltanto MetaEditor a 64 bit. Con più installazioni di MetaTrader usa **Browse...** per sceglierne una.
- **MQL5 root** deve puntare alla cartella dati dello stesso terminale (in MetaEditor: **File → Open Data Folder**, poi `MQL5`). Deve essere scrivibile perché la compilazione crea al suo interno una cartella temporanea.

Il controllo dimostra che il codice protetto compila nella configurazione utilizzata: **non sostituisce i test del comportamento in MetaTrader**.

8. Seed e build per singolo cliente

Ogni build utilizza un seed che determina la generazione di nomi e protezioni.

- **Campo seed vuoto:** viene scelto un nuovo seed casuale e ciascuna build è diversa.
- **Seed personalizzato:** a parità di progetto, preset e seed, l'output è riproducibile. È utile per ricreare una build consegnata.

- **Build per cliente:** utilizza un seed diverso per ogni cliente. Le copie saranno protette in modo differente, facilitando la distinzione fra le distribuzioni.

Conserva una corrispondenza privata fra cliente e seed: i seed sono informazioni riservate.

9. Build reversibili e Restore

Per impostazione predefinita le build **non sono reversibili**: l'output contiene soltanto codice protetto e non consente di recuperare l'originale. Conserva sempre il progetto originale in un backup o in un sistema di controllo versione.

Con **Reversible build** l'output include anche una copia cifrata e autenticata del progetto originale (AES-GCM, chiave derivata dal seed). Per le build reversibili è necessario un seed casuale sicuro: premi **Generate** per crearne uno da 256 bit. Il seed viene conservato anche nel vault locale privato (vedi sezione 14).

Procedura di ripristino:

1. Seleziona il `.mq5` principale protetto come file principale.
2. Scegli una cartella di output vuota per il progetto ripristinato.
3. Inserisci il seed utilizzato per quella build.
4. Premi **RESTORE**.

Il ripristino è esatto, byte per byte, compresi commenti, formattazione e codifica dei file. Seed errati o file modificati vengono rifiutati prima di scrivere qualsiasi dato.

> **Senza il seed una build reversibile non è ripristinabile, neppure da ALGOTS.** Proteggilo da perdite e accessi non autorizzati e non comunicarlo mai ai clienti.

Per le copie destinate ai clienti preferisci una build non reversibile.

10. Contenuto della cartella di output

- Il progetto protetto, pronto per essere compilato o distribuito. Con **Flatten includes**, il file principale può chiamarsi `YourEA_obfuscated.mq5`.
- `build_report.json`: report **pubblico** contenente stato, conteggi dei simboli trasformati e mantenuti, avvisi ed esito della compilazione. Non contiene nomi originali, mappa dei simboli, percorsi assoluti né seed.

I dati privati — mappa dei simboli (nomi originali ↔ nuovi), report diagnostico dettagliato e, nelle build reversibili, seed — risiedono nel vault locale sul PC, non nella cartella di output. Servono per assistenza interna e non devono essere distribuiti.

Regole della cartella di output. Deve essere vuota oppure già creata da ALGOTS Obfuscator. Ogni build la sostituisce integralmente e in sicurezza: i file rimossi dal progetto non rimangono dalle build precedenti; in caso di errore viene ripristinato il contenuto precedente. Le cartelle con dati estranei vengono rifiutate per evitare sovrascritture accidentali.

11. Interfaccia a riga di comando (CLI)

La CLI utilizza lo stesso motore e la stessa licenza dell'app desktop. È adatta a script, build in batch e pipeline CI.

Avvio

- Se durante l'installazione hai scelto **Add the CLI to the user PATH**, apri una **nuova** finestra PowerShell o Prompt dei comandi e digita `algots-obfuscator`.
- Altrimenti usa **ALGOTS Obfuscator CLI** nel menu Start oppure il percorso completo: `"%LOCALAPPDATA%\Programs\ALGOTS Obfuscator\CLI\ALGOTSObfuscatorCLI.exe"`.

Comandi

Comando	Scopo
`obfuscate MAIN -o OUTPUT`	Genera una copia protetta del progetto
`restore MAIN -o OUTPUT`	Ripristina l'originale da una build reversibile
`inspect MAIN`	Elenca sorgenti, include, risorse e dipendenze mancanti; non scrive file
`doctor [MAIN]`	Verifica ambiente, MetaEditor, root MQL5 e componenti
`wizard`	Procedura guidata interattiva con conferma finale
`license ...`	`activate`, `status`, `check`, `deactivate`, `forget-local`
`language [CODE]`	Elenca o imposta la lingua, condivisa con l'app desktop
`open-output PATH`	Apre la cartella di output in Explorer
`version`, `about`	Versione e informazioni sul prodotto

Tutti i comandi supportano `--help`.

Esempi eseguibili (PowerShell)

```
# 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
```

Opzioni Custom (ognuna disponibile anche nella forma `--no-…`): `--strip-comments`, `--flatten-includes`, `--minify`, `--encrypt-strings`, `--encode-constants`, `--opaque-if`, `--rename-parameters`, `--rename-locals`, `--rename-globals`, `--rename-members`, `--rename-inputs`.

Per l'automazione, `restore` accetta anche `--seed-file PATH`, file di testo UTF-8 contenente il seed. Evita `--seed` con un seed di ripristino sui computer condivisi: le righe di comando possono essere visibili nella cronologia della shell e negli elenchi dei processi.

Automazione

Opzione	Effetto
`--json`	Stampa esattamente un oggetto JSON su stdout, anche in caso di errore
`--quiet` / `-q`	Nasconde i riepiloghi testuali; avvisi ed errori rimangono su stderr
`--verbose` / `-v`	Mostra log del motore e diagnostica
`--color auto|always|never`	Colori del terminale
`--lang CODE`	Lingua del solo comando corrente, per esempio `--lang en`

I risultati JSON contengono sempre `schema_version`, `ok`, `command`, `exit_code` e `result` oppure `error`. Gli avvisi si trovano in `result.warnings`.

Codici di uscita

Codice	Significato
0	Operazione completata; possono esserci avvisi
1	Errore interno inatteso
2	Argomenti della riga di comando non validi
3	Input, configurazione, file system o cartella di output non validi
4	Percorso toolchain/ambiente non valido o root MQL5 ambigua
5	Controllo di compilazione MetaEditor fallito o scaduto
6	Restore rifiutato: seed errato, file modificato o destinazione non valida
10	Attivazione licenza richiesta
11	Verifica online della licenza richiesta
20	Licenza rifiutata: scaduta, disabilitata o non valida per il prodotto
30	Errore del servizio licenze; riprovare più tardi
40	Impossibile accedere ai dati locali della licenza
50	Problema di configurazione del prodotto; contattare l'assistenza
51	Controllo integrità pacchetto fallito; reinstallare
130	Operazione interrotta (Ctrl+C)

12. Gestione della licenza

Nell'app desktop premi **License**; nella CLI utilizza `license`.

Azione	App desktop	CLI
Attivare	**Activate License**	`license activate`
Vedere lo stato	Visualizzato nella finestra	`license status`
Forzare il controllo online	**Check online**	`license check`
Liberare il dispositivo	**Deactivate This Device**	`license deactivate`

Azione	App desktop	CLI
Eliminare soltanto i dati locali	—	`license forget-local --yes`

- **Passaggio a un nuovo PC:** disattiva la licenza sul vecchio computer e poi attivala sul nuovo.
- `forget-local` elimina soltanto i dati della licenza sul PC, **non** libera l'attivazione sul server; per farlo usa `deactivate`.
- La licenza è verificata periodicamente online. Se il server è irraggiungibile puoi lavorare per un periodo limitato dall'ultimo controllo riuscito; la finestra mostra **Offline use allowed until**. Alla scadenza connettiti a Internet e premi **Check online**.
- Riportare indietro l'orologio di sistema non prolunga l'uso offline.

Negli ambienti automatizzati `license activate --key-stdin` legge la chiave dallo standard input; è supportata anche la variabile d'ambiente `MQL5OBF_LICENSE_KEY`.

13. Lingue

App desktop, CLI e installer sono disponibili in inglese, italiano, spagnolo, francese, tedesco, portoghese, turco, russo, cinese semplificato, giapponese, coreano, arabo e persiano.

Seleziona la lingua nell'app mediante **Language**. La CLI usa la stessa scelta; altrimenti segue la lingua di Windows e, se non supportata, usa l'inglese.

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

Nomi dei comandi, opzioni, codici d'uscita e chiavi JSON non vengono mai tradotti.

14. File e cartelle sul PC

Percorso	Contenuto
`%LOCALAPPDATA%\Programs\ALGOTS Obfuscator`	Applicazione
`%APPDATA%\MQL5Obfuscator\settings.json`	Preferenze
`%LOCALAPPDATA%\MQL5Obfuscator\Licensing`	Stato licenza cifrato con Windows DPAPI per l'account utente
`%LOCALAPPDATA%\MQL5Obfuscator\Builds`	Dati privati: mappe dei simboli e report diagnostici

Percorso	Contenuto
`%LOCALAPPDATA%\MQL5Obfuscator\Recovery`	Seed e cronologia delle build reversibili

Le cartelle `Builds` e `Recovery` sono riservate. Esegui backup se dipendi da esse per supporto o ripristino; **non inviarle mai ai clienti**.

15. Buone pratiche

- Conserva i sorgenti originali in controllo versione o backup: ALGOTS Obfuscator non è uno strumento di backup.
- Inizia ogni nuovo progetto con **Safe**, poi passa a **Harden**.
- Mantieni attivo il compile check e verifica il programma protetto in MetaTrader prima della consegna.
- Distribuisci ai clienti build non reversibili.
- Usa seed differenti per cliente e registra privatamente ogni associazione.
- Non distribuire vault privato, mappe dei simboli o seed.
- L'offuscamento aumenta lo sforzo richiesto per leggere e riutilizzare il codice, ma non rende impossibile il reverse engineering: per codice di valore, affianca licenze e contratti adeguati.

16. Risoluzione dei problemi

I pulsanti OBFUSCATE e RESTORE sono disabilitati. La licenza non è attiva o richiede verifica online. Premi **License**, poi **Activate License** o **Check online**.

«MetaEditor was not found» / «select the terminal MQL5 root». Premi **Auto** oppure seleziona manualmente `metaeditor64.exe` e la cartella `MQL5` del terminale. Puoi anche disattivare il controllo di compilazione automatico.

La build fallisce con errori di compilazione. Verifica anzitutto che il progetto originale compili senza errori nello stesso MetaEditor. Prova quindi Safe. Se Safe compila e Harden no, contatta l'assistenza allegando soltanto il report pubblico `build_report.json` (mai il vault privato).

File `#include <...>` non trovati. Imposta la root MQL5 del terminale che contiene i file richiesti.

Cartella di output rifiutata. Scegli una cartella vuota o creata in precedenza da ALGOTS Obfuscator.

La build si interrompe a causa di `__PATH__`. `__PATH__` includerebbe nel codice distribuito il percorso assoluto del file sorgente, eventualmente con il nome utente. Rimuovilo o abilita **Allow absolute `__PATH__`** soltanto se accetti questa divulgazione.

«Package integrity check failed». Reinstalla usando l'installer. Se un antivirus ha messo in quarantena un file, contatta support@algots.net per una diagnosi e verifica origine e integrità dei file prima di agire. Non aggirare le protezioni.

`algots-obfuscator` **non è riconosciuto.** Apri un nuovo terminale dopo l'installazione; se il problema persiste, ripeti il setup selezionando **Add the CLI to the user PATH**.

Licenza: «activation limit reached». Disattiva la licenza su un dispositivo non più utilizzato oppure contatta l'assistenza.

17. Disinstallazione

Vai in **Settings → Apps → Installed apps → ALGOTS Obfuscator → Uninstall**, oppure usa **Uninstall ALGOTS Obfuscator** nel menu Start. La disinstallazione elimina anche la cartella CLI dal PATH del tuo utente.

Per liberare la licenza per un altro PC, **disattivala prima di disinstallare**. Impostazioni e vault privato sono conservati: se desideri eliminarli, cancella anche le cartelle `MQL5Obfuscator` indicate nella sezione 14.

18. Assistenza

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

Comunica la versione (`algots-obfuscator version`), la versione di Windows e, per problemi di build, il solo `build_report.json` pubblico. Non inviare mai seed, mappe dei simboli o contenuto del vault privato.

MetaTrader e MQL5 sono marchi di MetaQuotes Ltd. ALGOTS Obfuscator è un prodotto indipendente e non è affiliato a MetaQuotes.