

XLS PADLOCK

Guida utente di XLS Padlock

Protezione, licenze e distribuzione di cartelle di lavoro
Excel

Version 2026.0

G.D.G. Software

www.xlspadlock.com

Copyright © G.D.G. Software 2018-2026. Tutti i diritti riservati.

Indice

01 Informazioni su questa guida

02 Introduzione a XLS Padlock

03 Scaricare e installare

04 Impostazioni dell'applicazione

05 XLS Padlock Manager

06 Come proteggere una cartella di lavoro Excel

07 Sicurezza

08 Opzioni cloud e IA

09 Migliorare la protezione

10 Controllo di accesso alla cartella di lavoro

11 Restrizioni della cartella di lavoro Excel

12 Protezione con password dei fogli

13 Combinare la protezione di Excel e di XLS Padlock

14 Protezione delle formule: Excel rispetto a XLS Padlock

15 Proteggere le formule con XLS Padlock

16 Contrassegnare le celle per la protezione delle formule

17 Metodo di protezione delle formule

- [18 Disattivare la protezione delle formule](#)

- [19 Informazioni sul VBA Compiler integrato](#)

- [20 Protezione reale del codice VBA](#)

- [21 Riferimento sintassi](#)

- [22 Ricettario API VBA](#)

- [23 Scrivere e compilare VBA sicuro](#)

- [24 Richiamare il VBA compilato in fase di esecuzione](#)

- [25 Accedere agli oggetti Excel](#)

- [26 Passare array](#)

- [27 Passare più parametri](#)

- [28 Gestire gli errori](#)

- [29 OLE error 800A03EC](#)

- [30 Protezione del codice VBA](#)

- [31 Nascondere e bloccare il codice VBA](#)

- [32 Bloccare il progetto VBA](#)

- [33 Vietare l'accesso al VBE](#)

- [34 Guida all'attivazione e alle licenze](#)

- [35 Chiavi di attivazione](#)

- [36 Configurare le chiavi di attivazione](#)

- [37](#) Chiavi di attivazione con blocco hardware

- [38](#) Attivazione online

- [39](#) Convalida online

- [40](#) Editor del modulo di registrazione

- [41](#) Mostrare il contratto di licenza

- [42](#) Generatore di chiavi (portatile e server)

- [43](#) SDK del generatore di chiavi

- [44](#) Generatore di chiavi autonomo

- [45](#) Restrizioni sulle chiavi

- [46](#) Protezione tramite USB o dongle

- [47](#) Disattivazione

- [48](#) Creare cartelle di lavoro di prova

- [49](#) Chiudere dopo un tempo prestabilito

- [50](#) Verificare lo stato di prova

- [51](#) Giorni di prova rimanenti

- [52](#) Opzioni di salvataggio della cartella di lavoro

- [53](#) Salvataggio e caricamento

- [54](#) Modalità di salvataggio: completo o valori delle celle

- [55](#) Definire le celle da salvare e ripristinare

- 56 Ripristinare e salvare con VBA

- 57 Accedere alla cartella di lavoro protetta e ai file complementari

- 58 Aggiungere file complementari

- 59 Cartella di archiviazione dei salvataggi

- 60 Archiviare le modifiche all'interno dell'EXE

- 61 Limitare il caricamento e il salvataggio

- 62 Aprire e decifrare i salvataggi

- 63 Bloccare i salvataggi su una macchina

- 64 Riferimenti esterni e collegamenti ipertestuali

- 65 Ottenere il percorso accanto alla cartella di lavoro compilata

- 66 Personalizzare l'applicazione

- 67 Pacchettizzazione dell'applicazione

- 68 Schermata iniziale

- 69 Cambiare l'icona dell'EXE

- 70 Finestra di Excel all'avvio

- 71 Eseguire come applicazione solo VBA

- 72 Informazioni sulla versione dell'EXE

- 73 Opzioni da riga di comando

- 74 Nascondere la finestra di caricamento

- 75 Localizzazione e traduzione

- 76 Opzioni avanzate

- 77 Ignorare gli errori di elaborazione

- 78 Disattivare le informazioni di debug

- 79 Interfaccia personalizzata di Excel

- 80 Componenti aggiuntivi di Excel

- 81 Distribuire l'applicazione

- 82 Distribuire una cartella di lavoro protetta

- 83 Firmare digitalmente l'EXE

- 84 Creare un programma di installazione

- 85 Compilare l'EXE in base all'architettura di Excel

- 86 Versioni di Excel

- 87 Aggiornamenti della cartella di lavoro

- 88 Aggiornamenti web automatici

- 89 Migrare i dati utente tra gli aggiornamenti

- 90 Salvare e ripristinare le impostazioni tramite modelli

- 91 Errore del registro o EREGISTRYEXCEPTION

- 92 Violazione di accesso all'indirizzo

- 93 Errore "Failed to set data for 'Data'"

94 Perché l'EXE è così grande

95 Ripristinare il file XLS dall'EXE

96 Link di assistenza

Informazioni su questa guida

Copyright © G.D.G. Software 2013-2026

Tutti i diritti riservati. Nessuna parte di quest'opera può essere riprodotta in alcuna forma o con alcun mezzo, grafico, elettronico o meccanico, inclusi la fotocopiatura, la registrazione, l'archiviazione su nastro o i sistemi di memorizzazione e recupero delle informazioni, senza il permesso scritto dell'editore.

Ai clienti titolari di una licenza XLS Padlock è concesso il permesso di stampare questa guida per uso privato o didattico.

Microsoft Excel® e Office® sono marchi registrati di Microsoft Corporation.

Qualsiasi prodotto menzionato in questo documento può essere un marchio commerciale e/o un marchio registrato dei rispettivi proprietari. L'editore e l'autore non rivendicano alcun diritto su tali marchi.

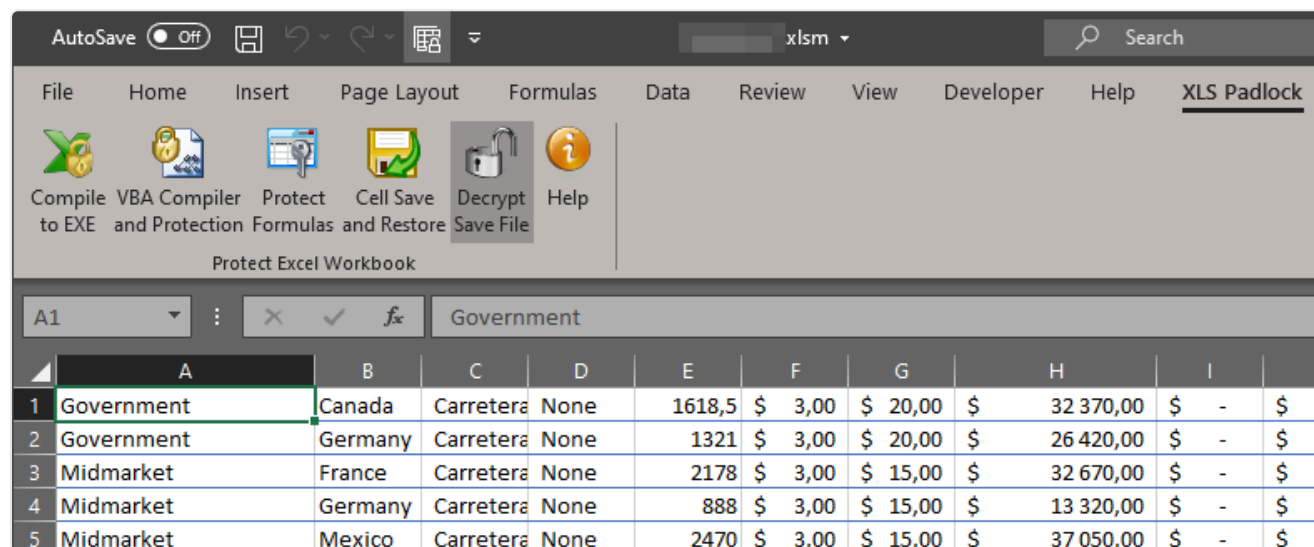
Sebbene sia stata adottata ogni precauzione nella preparazione di questo documento, l'editore e l'autore non si assumono alcuna responsabilità per errori od omissioni, né per danni derivanti dall'uso delle informazioni in esso contenute o dall'uso dei programmi e del codice sorgente che lo accompagnano. In nessun caso l'editore o l'autore potranno essere ritenuti responsabili per qualsiasi perdita di profitti o per qualsiasi altro danno commerciale causato o presumibilmente causato direttamente o indirettamente da questo documento.

Prima edizione: gennaio 2013. Aggiornato: giugno 2026.

Versione 2026.0, [Leggi il registro delle modifiche](#)

Introduzione a XLS Padlock

XLS Padlock è una potente soluzione di protezione e gestione delle licenze per Microsoft Excel, progettata per proteggere le cartelle di lavoro dalla copia. Funziona come un compilatore, consentendo di convertire le cartelle di lavoro Excel in applicazioni autonome e sicure.



Questo consente di distribuire i file Excel in tutta sicurezza, di controllare l'accesso degli utenti e di impedire la copia dei fogli di calcolo. È possibile limitare funzionalità chiave di Excel, tra cui:

- Il menu contestuale dei fogli di calcolo (clic destro)
- La funzione copia/incolla
- La barra della formula
- Il salvataggio e la stampa
- L'accesso al progetto VBA

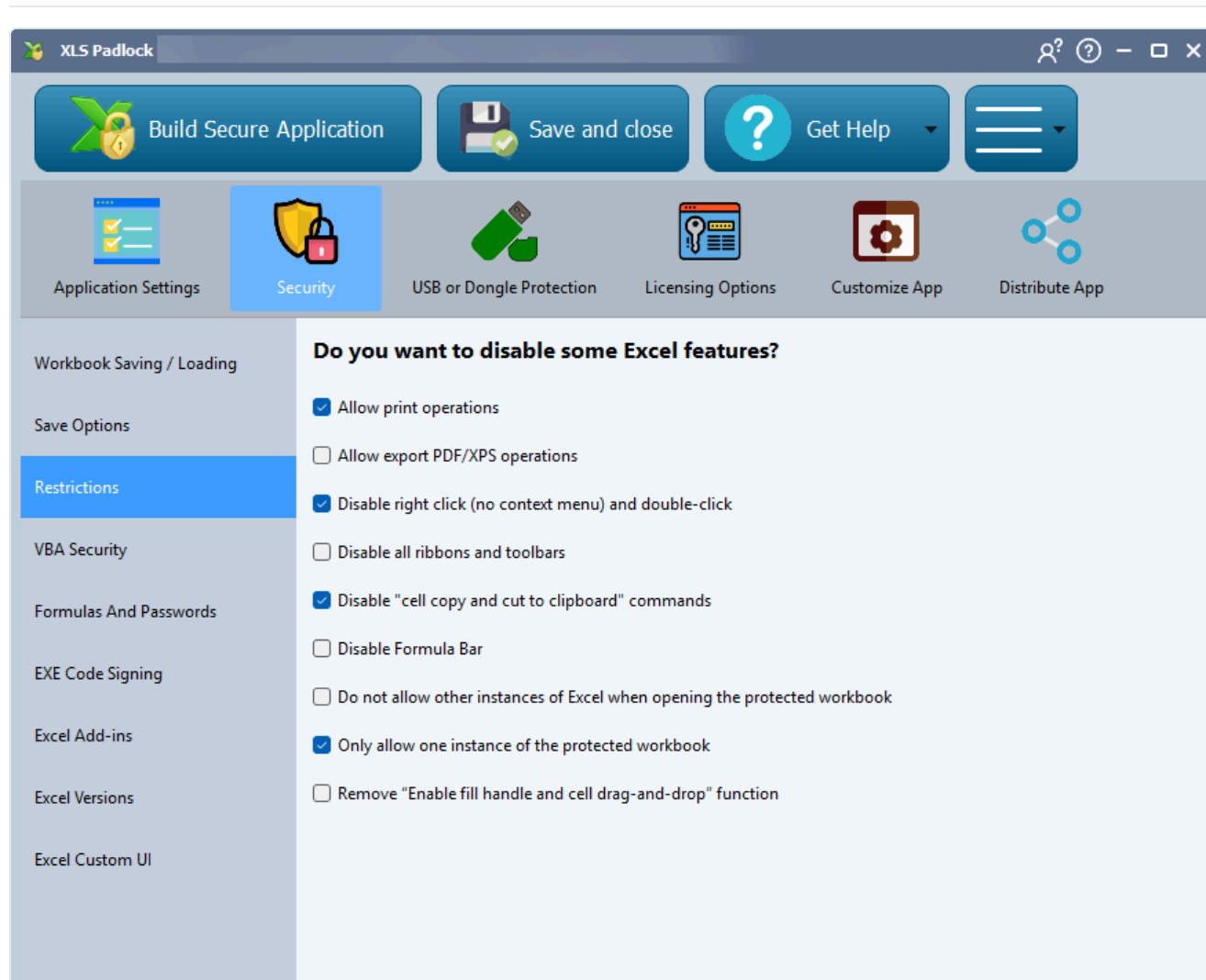
XLS Padlock offre diversi modi per proteggere i file, ad esempio **bloccandoli su una specifica chiavetta USB**, utilizzando una **robusta protezione tramite dongle** oppure emettendo **Activation Keys con blocco hardware**. Si ottiene inoltre un controllo granulare sulla protezione delle formule, che consente agli utenti di usare le formule senza poterle visualizzare né copiare.

Protezione avanzata del codice VBA

Grazie al VBA Compiler integrato, XLS Padlock consente di **compilare il codice VBA sensibile in byte-code sicuro, rendendolo inaccessibile agli utenti finali**. Le macro VBA sono realmente protette, perché il codice sorgente originale viene rimosso.

Il compilatore **non è un semplice offuscatore**; trasforma il codice VBA in codice binario e lo memorizza in modo sicuro all'interno dell'EXE dell'applicazione. Per completare la protezione, è possibile bloccare il progetto VBA, **rendendo inutili gli strumenti di cracking delle password**. [Scopri di più sulla protezione del codice VBA](#).

Opzioni di sicurezza complete



Le applicazioni possono essere configurate per **scadere dopo un numero di giorni prestabilito**, dopo un determinato numero di utilizzi oppure a una data fissa. È possibile visualizzare una schermata di promemoria personalizzabile per le versioni di prova. **L'attivazione online** automatizza il processo di gestione delle licenze, offrendo un **controllo remoto** sulle persone autorizzate ad accedere alle cartelle di lavoro.

Le applicazioni di cartella di lavoro protetta sono autonome e richiedono soltanto Microsoft Excel per funzionare. Tutte le funzionalità di Excel sono pienamente supportate.

👉 È inoltre possibile personalizzare e brandizzare le applicazioni:

- Aggiungere un'icona personalizzata e informazioni sul copyright.
- Tradurre tutto il testo visibile all'utente in qualsiasi lingua.
- Visualizzare un contratto di licenza (EULA) o una schermata iniziale all'avvio.
- Firmare digitalmente le cartelle di lavoro compilate con Authenticode.
- Creare un programma di installazione professionale per la distribuzione.
- Avvisare gli utenti delle nuove versioni grazie alla funzionalità integrata di aggiornamento web.

Compatibile con Office 365 ed Excel 2024, 2021, 2019, 2016, 2013, 2010, 2007 (SP3), 2002, sia nelle versioni a 32 bit sia a 64 bit.

Avvertenza

Per creare applicazioni a 64 bit è necessaria una versione a 64 bit di Windows.

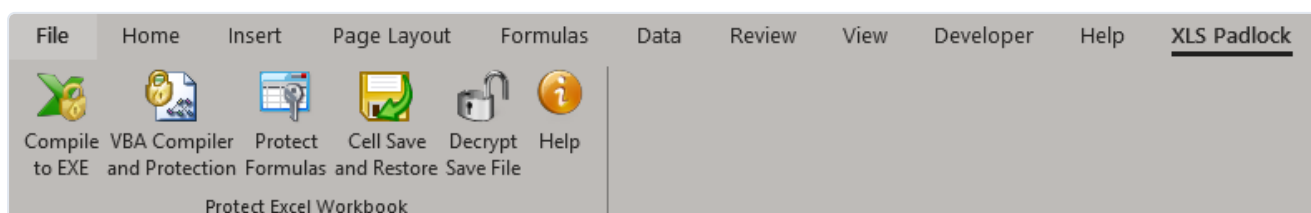
👉 Da scoprire subito:

- [Scaricare e installare XLS Padlock](#)
- [Come proteggere una cartella di lavoro Excel](#)
- [Sito web di XLS Padlock](#)
- [Novità](#)

Scaricare e installare

Sul nostro sito web all'indirizzo <https://www.xlspadlock.com/download>, è necessario selezionare il programma di installazione corrispondente alla propria versione di Microsoft Excel (32 bit o 64 bit). Il programma di installazione verificherà che si stia installando la versione corretta.

Dopo un'installazione riuscita, **XLS Padlock è integrato direttamente in Excel**, aparendo come una nuova scheda nella barra multifunzione. Per utilizzare XLS Padlock, è necessario prima aprire la propria cartella di lavoro Excel.

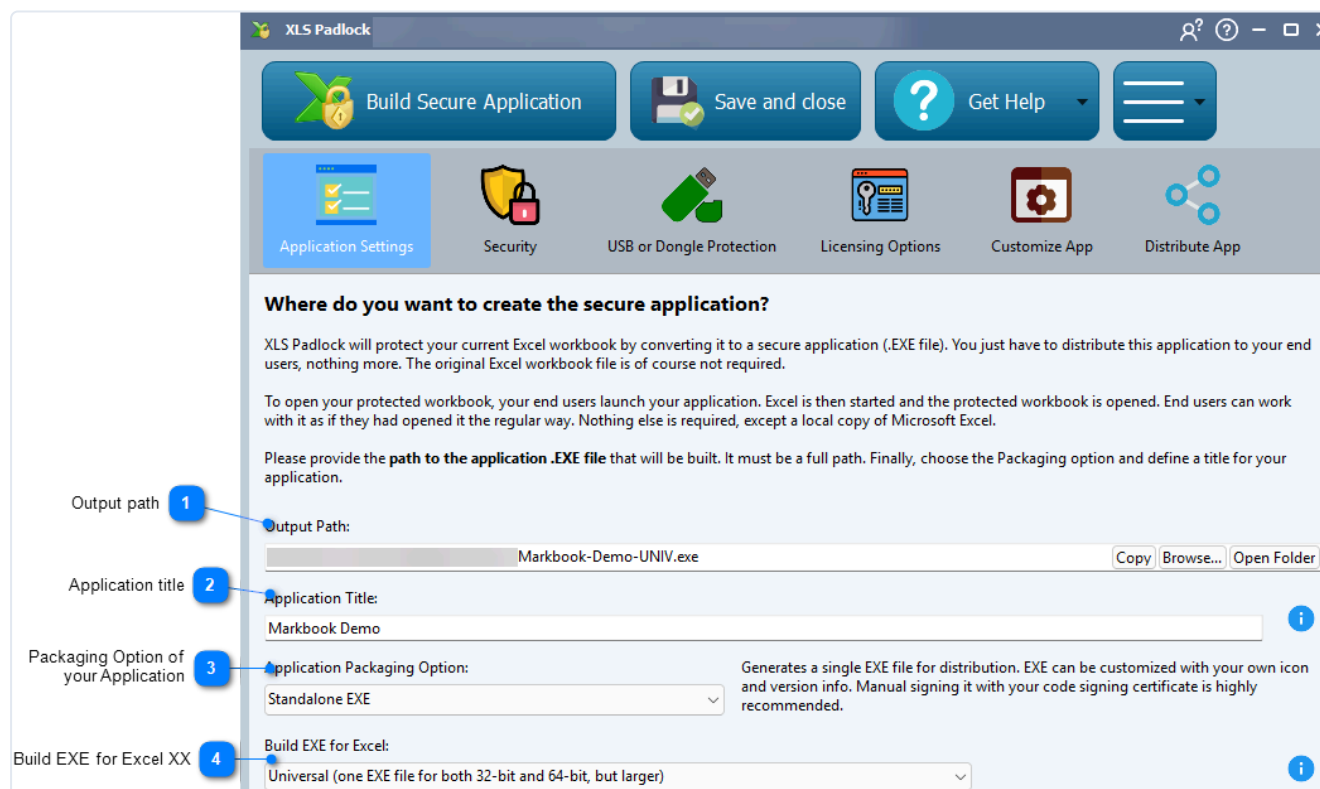


Sul desktop vengono creati anche due collegamenti: uno per questa guida utente e l'altro per il XLS Padlock Manager.

- [XLS Padlock Manager](#)
- [Come proteggere una cartella di lavoro Excel](#)

Impostazioni dell'applicazione

La pagina Application Settings contiene le impostazioni obbligatorie del progetto.



Impostazioni principali

Output Path

☞ Definisce la posizione in cui verrà creato il file EXE finale.

XLS Padlock compila la cartella di lavoro in un unico file eseguibile (EXE). Nel campo "Output Path" è necessario specificare il percorso completo e il nome del file in cui si desidera creare l'applicazione.

SUGGERIMENTO

Non dimenticare di includere l'estensione `.exe` nel nome del file.

Sono accettati anche i percorsi relativi; questi saranno relativi alla cartella che contiene il file della cartella di lavoro di origine. Per impostazione predefinita, XLS Padlock propone un percorso nella stessa cartella della cartella di lavoro di origine, ma è possibile sostituirlo con qualsiasi posizione.

Application Title

☞ Imposta il titolo della finestra principale dell'applicazione.

È possibile impostare un titolo personalizzato per l'applicazione. Questo titolo viene visualizzato nella barra del titolo della finestra, sostituendo il testo predefinito "Microsoft Excel".

Se si desidera visualizzare in modo dinamico il nome di un file di salvataggio caricato, è possibile utilizzare i seguenti segnaposto nel titolo:

- `%SAVEFILENAME%` : Sostituito dal solo nome del file (ad esempio, `MyData.xlsx`).
- `%SAVEFULLNAME%` : Sostituito dal percorso completo del file di salvataggio (ad esempio, `C:\Users\Me\Documents\MyData.xlsx`).

Ad esempio, un titolo impostato su `My Application, %SAVEFILENAME%` verrebbe visualizzato come "My Application, MyData.xlsx" quando l'utente ha quel file aperto.

NOTA

Se viene caricata la cartella di lavoro originale, non salvata, i segnaposto saranno vuoti.

Application Packaging Option

Vedere [Packaging Option](#): Scegliere tra un EXE autonomo o un bundle per la distribuzione.

Build EXE for Excel

Vedere [Build EXE for Excel](#): Indicare se compilare per la compatibilità a 32 bit, 64 bit o universale.

Configurazione aggiuntiva

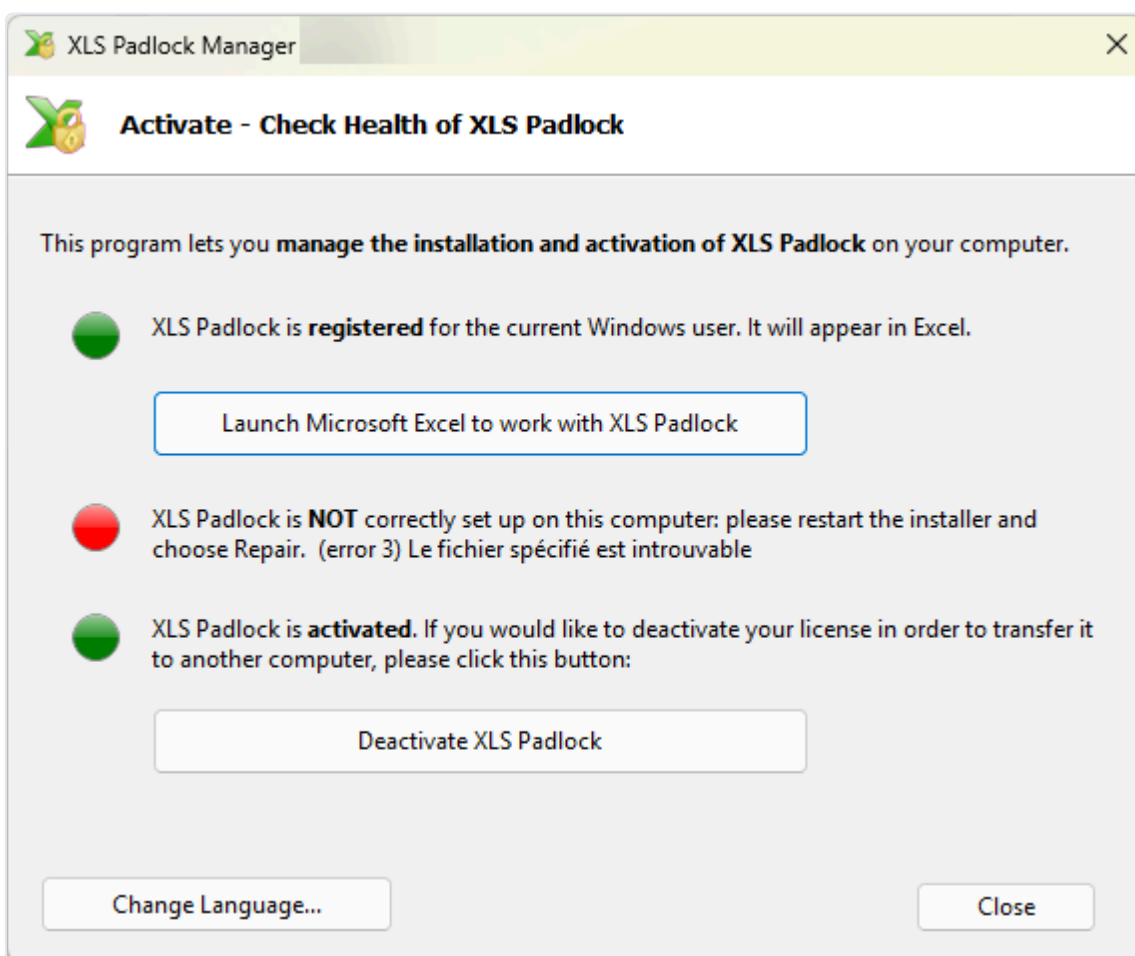
- [Configurare le opzioni avanzate](#)
- [Aggiungere file complementari](#)

XLS Padlock Manager

XLS Padlock Manager è un'applicazione autonoma che consente di **verificare se XLS Padlock è configurato e registrato correttamente per l'utente Windows corrente**.

XLS Padlock **deve essere registrato per ogni utente Windows che desidera utilizzarlo**. Quando si installa XLS Padlock su un computer, di norma viene registrato solo per l'account amministratore che ha eseguito l'installazione. Gli altri account utente devono utilizzare XLS Padlock Manager per registrare il software.

- Per registrare XLS Padlock per il proprio account utente Windows, avviare "XLS Padlock for Excel, Manager" dal desktop di Windows. Apparirà la finestra seguente:



- Se tutti gli indicatori sono verdi, l'installazione è corretta.
- Se il primo indicatore è rosso, fare clic su "**Enable XLS Padlock for the current Windows user**" (Abilita XLS Padlock per l'utente Windows corrente). XLS Padlock verrà quindi registrato e sarà possibile iniziare a utilizzarlo. Questa operazione non richiede diritti di amministratore.

**NOTA**

Se si dispone di una licenza XLS Padlock, è possibile utilizzare il manager anche per attivarla.

👉 [Come proteggere una cartella di lavoro Excel](#)

Come proteggere una cartella di lavoro Excel

Tutorial video disponibile

Scopri questi tutorial video su XLS Padlock realizzati da un esperto di Excel:

<https://excelvbaisfun.com/?ref=5>

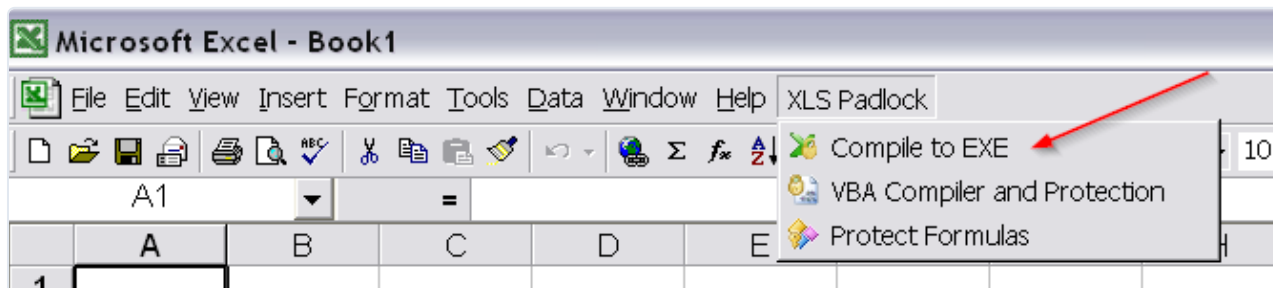
1\.. Aprire e salvare la cartella di lavoro

Iniziare aprendo in Excel la cartella di lavoro che si desidera proteggere. Assicurarsi che tutte le modifiche siano salvate prima di procedere.

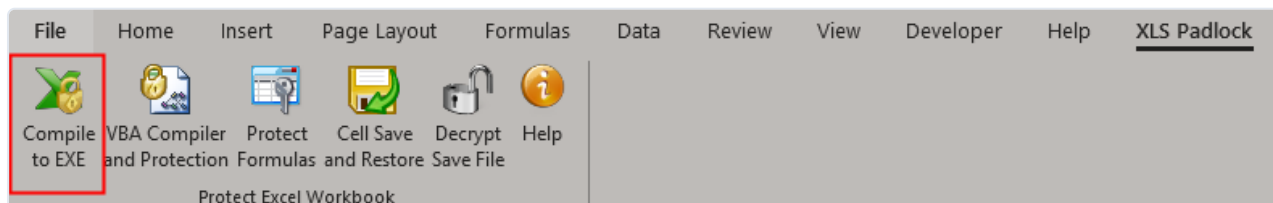
2. Configurare le opzioni di protezione

Per iniziare, aprire l'interfaccia di XLS Padlock dalla barra multifunzione di Excel:

- **Prima di Excel 2007:** utilizzare il menu "XLS Padlock".



- **Excel 2007 e versioni successive:** fare clic su "Secure Compile to Exe" nella scheda "XLS Padlock" della barra multifunzione.



NOTA

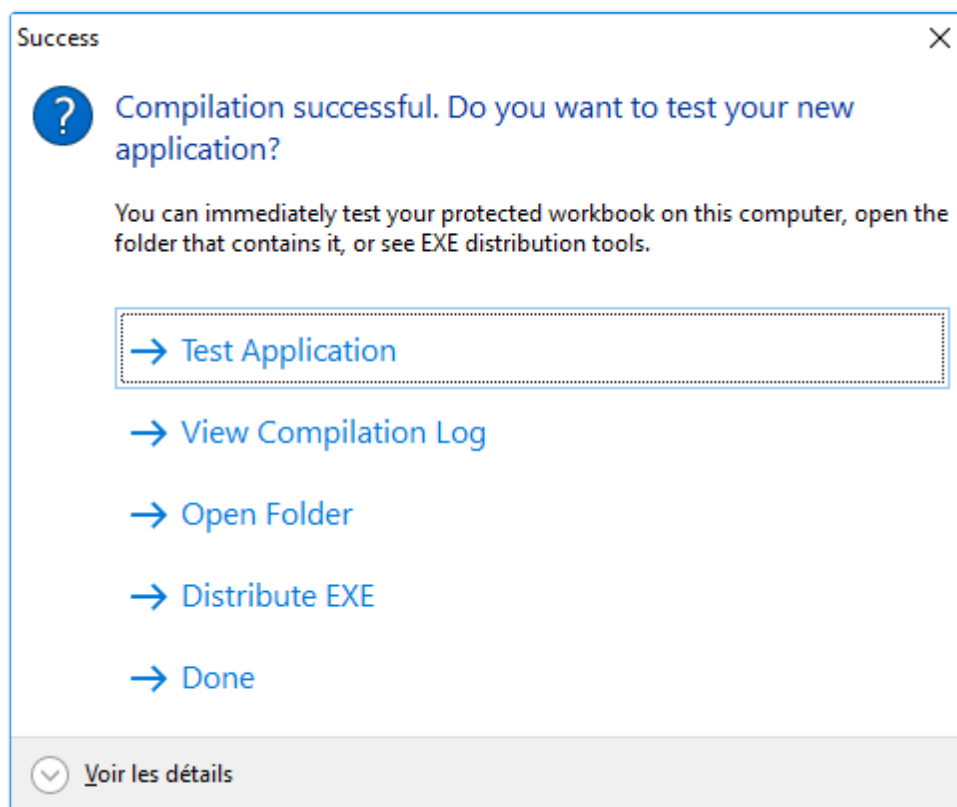
Salvare sempre la propria cartella di lavoro Excel prima di aprire l'interfaccia di XLS Padlock, affinché tutte le modifiche recenti siano incluse nella compilazione.

3. Generare l'applicazione

Dopo aver configurato le impostazioni della propria applicazione, fare clic sul pulsante "Build Secure Application" oppure premere **F5**. XLS Padlock compilerà la cartella di lavoro e creerà il file EXE dell'applicazione.



Al completamento, apparirà un messaggio di conferma:



Da questa finestra di dialogo è possibile eseguire immediatamente l'applicazione, consultare il registro di compilazione (particolarmente utile per risolvere gli errori) oppure aprire la cartella di destinazione.

SUGGERIMENTO

XLS Padlock non modifica il file della cartella di lavoro Excel originale durante la compilazione. Conservare sempre una copia di backup della cartella di lavoro originale.

Risoluzione dei problemi di formattazione

Se la cartella di lavoro protetta mostra risultati strani o problemi di formattazione, provare ad attivare l'opzione "[Use Excel automation for formula protection](#)" nella pagina "Formulas and Passwords".

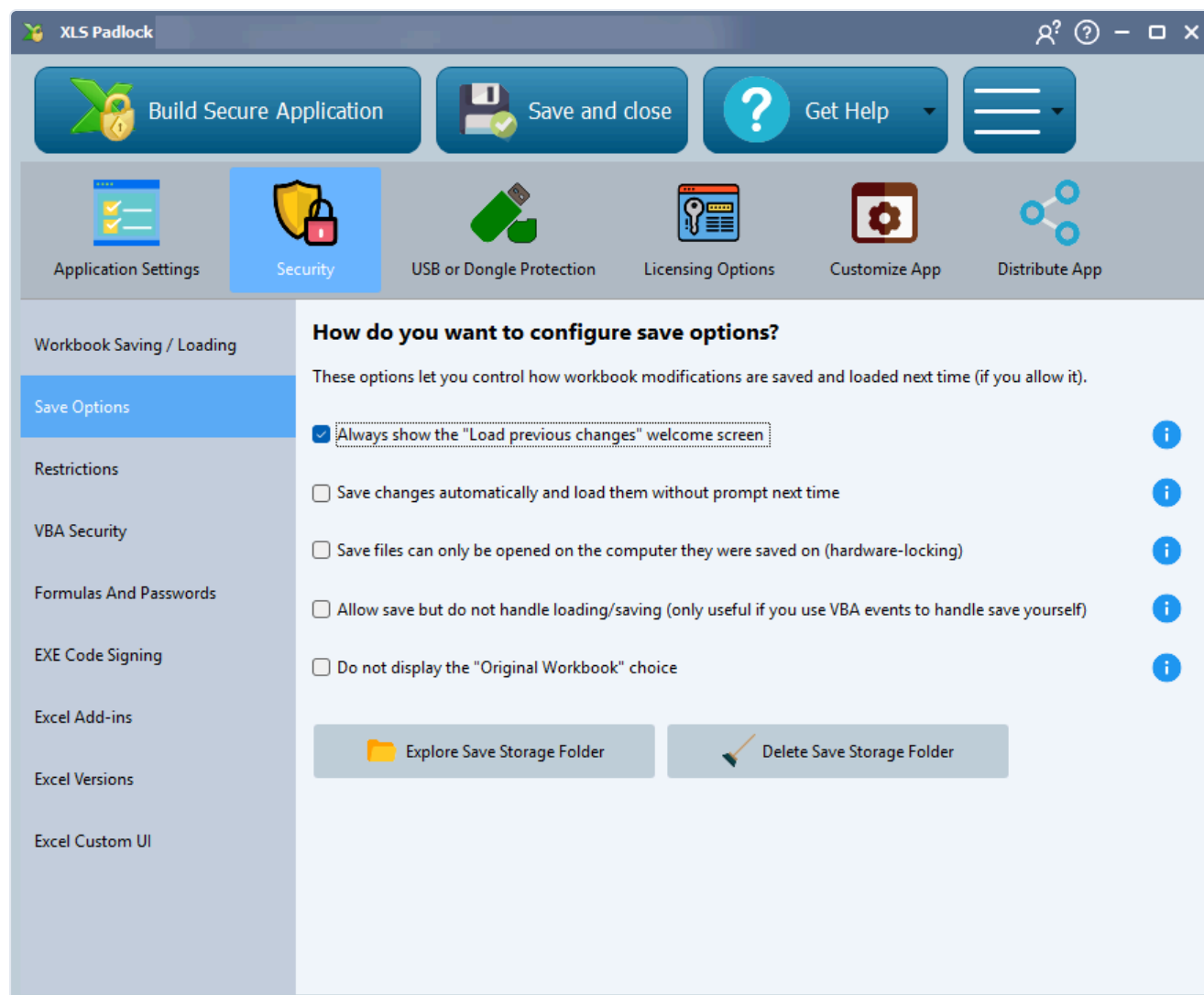
👉 Vedere anche:

[Come distribuire una cartella di lavoro protetta](#)

Sicurezza

XLS Padlock offre molti modi per proteggere la cartella di lavoro Excel, tra cui opzioni di salvataggio, restrizioni sulle funzionalità e condizioni per l'apertura dell'applicazione, come la richiesta di una [chiave di attivazione](#) o la [presenza di un dongle o di una chiavetta USB](#).

Questo argomento fornisce una panoramica delle opzioni di sicurezza disponibili in "**Security**" in XLS Padlock.



Workbook Saving / Loading

Consente di configurare se gli utenti finali possono salvare le modifiche apportate alla cartella di lavoro protetta. Quando il salvataggio è abilitato, XLS Padlock crea file sicuri `.XLSC` o `.XLSCE` che possono essere aperti solo dalla propria applicazione, proteggendo così i dati dell'utente.

[→ Scopri di più sul salvataggio e il caricamento della cartella di lavoro](#)

Save Options

Consente di perfezionare il comportamento del salvataggio con opzioni come il salvataggio automatico delle modifiche alla chiusura, il blocco hardware dei file di salvataggio su un computer specifico oppure la visualizzazione permanente della schermata di benvenuto, per permettere agli utenti di scegliere se ripartire da zero o caricare un lavoro precedente.

[→ Esplora le opzioni di salvataggio avanzate](#)

Restrictions

Consente di limitare le funzionalità chiave di Excel per rafforzare la sicurezza. È possibile disabilitare il menu contestuale del clic destro, la barra della formula, i comandi di copia/incolla, la stampa e molto altro, per impedire agli utenti di accedere alle parti sensibili della cartella di lavoro o di modificarle.

[→ Vedi le funzionalità di restrizione](#)

VBA Security

Consente di proteggere il codice VBA da qualsiasi consultazione o modifica. È possibile bloccare il progetto VBA con una password nascosta all'utente oppure bloccare completamente l'accesso all'editor VBA (VBE), rendendo inutili gli strumenti di violazione delle password. Per una protezione ottimale, utilizzare il VBA Compiler.

[→ Vedi i diversi mezzi di protezione VBA](#)

Formulas and Passwords

Consente di proteggere le formule e la struttura dei fogli di calcolo. Utilizzare la protezione delle formule di XLS Padlock per permettere agli utenti di interagire con le formule senza poterle vedere né copiare. È inoltre possibile gestire le password della cartella di lavoro e dei fogli di calcolo direttamente dall'interfaccia.

[→ Scopri di più sulla protezione delle formule](#)

EXE Code Signing

Consente di firmare digitalmente il file `.EXE` compilato per ispirare fiducia agli utenti ed evitare gli avvisi di Windows SmartScreen e dei software antivirus. XLS Padlock può automatizzare il processo di firma se si dispone di un certificato di firma del codice.

[→ Come firmare digitalmente l'applicazione](#)

Excel Add-ins

Per impostazione predefinita, XLS Padlock disabilita la maggior parte dei componenti aggiuntivi per motivi di sicurezza. Questa sezione consente di riabilitare in modo selettivo specifici componenti aggiuntivi COM o componenti aggiuntivi Excel comuni (come Analysis ToolPak o Solver) necessari al corretto funzionamento della cartella di lavoro.

[→ Come gestire i componenti aggiuntivi di Excel](#)

Excel Versions

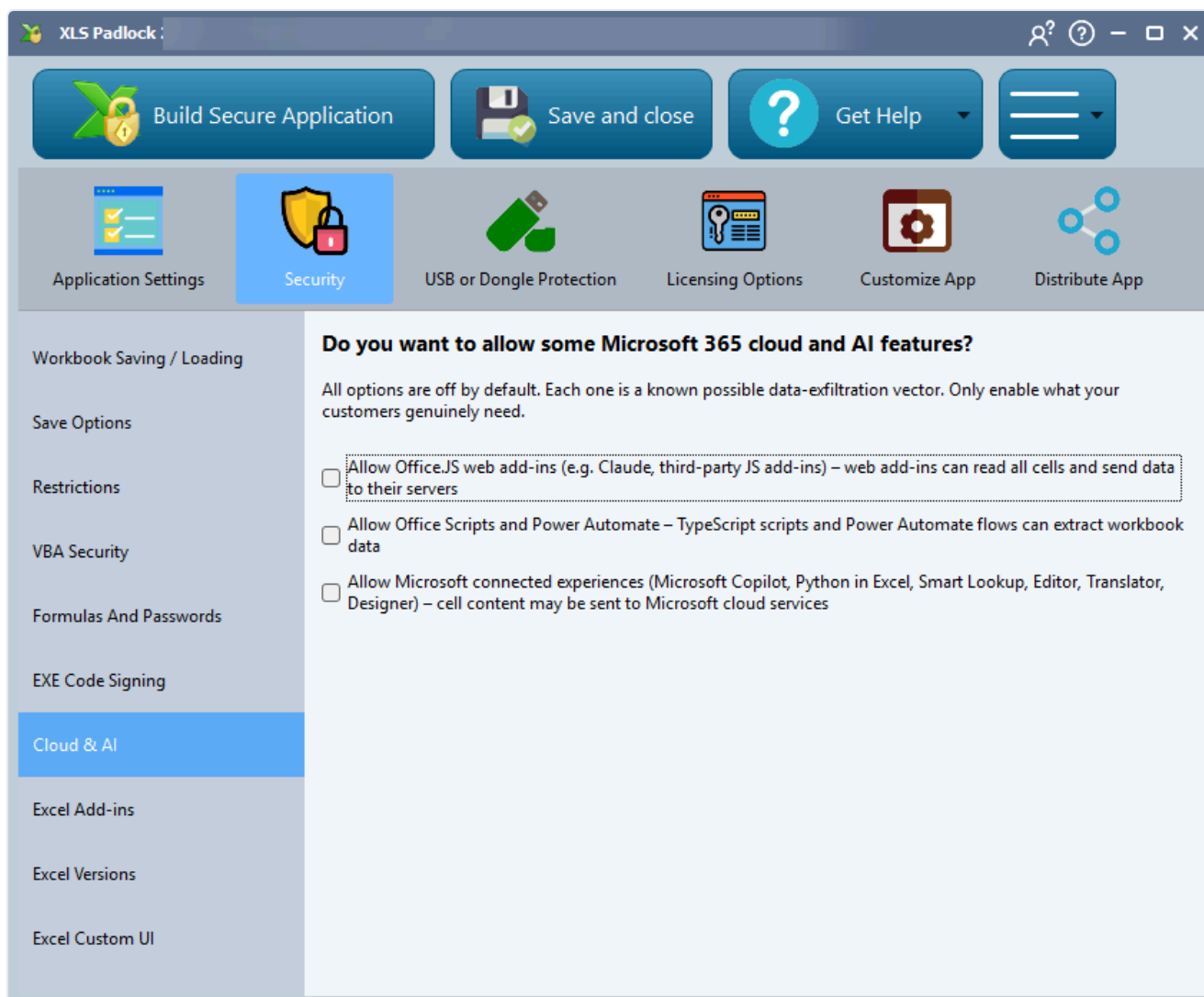
Consente di specificare quali versioni di Microsoft Excel sono compatibili con l'applicazione protetta. Se un utente tenta di eseguire l'EXE su una versione non supportata, verrà visualizzato un messaggio di errore personalizzabile.

[→ Come richiedere versioni specifiche di Excel](#)

Opzioni cloud e IA

Excel moderno include diverse funzionalità cloud e IA in grado di leggere la cartella di lavoro e inviargli il contenuto a server esterni. Per mantenere ermetica l'applicazione protetta, XLS Padlock le blocca tutte per impostazione predefinita. È l'utente a decidere, una per una, quali consentire.

Questi parametri si trovano nel designer di XLS Padlock, sotto la scheda **Security** (Sicurezza), nella pagina **Cloud & AI** (Cloud e IA).



Perché queste funzionalità sono bloccate per impostazione predefinita

Ogni opzione di questa pagina è una via nota attraverso cui i dati della cartella di lavoro possono lasciare la macchina dell'utente finale. Una cartella di lavoro protetta è pensata per mantenere le formule, i dati e il codice VBA sotto il proprio controllo, perciò XLS Padlock parte dalla posizione più sicura: tutto disattivato. Attivare solo ciò di cui i clienti hanno realmente bisogno.

Opzioni disponibili

Tutte e tre le opzioni sono deselezionate (bloccate) per impostazione predefinita.

Allow Office JS web add-ins

I componenti aggiuntivi web Office JS (chiamati anche web add-in) vengono eseguiti all'interno di Excel e possono leggere ogni cella della cartella di lavoro aperta e inviare tali dati ai propri server. Questo include componenti aggiuntivi di terze parti come il plugin **Claude for Excel** e qualsiasi altro componente aggiuntivo JS installato dall'Office store o caricato in modalità indipendente (sideloaded).

Quando questa opzione è disattivata, i componenti aggiuntivi web non possono caricarsi nella cartella di lavoro protetta. Abilitarla solo se l'applicazione dipende da uno specifico componente aggiuntivo web di cui ci si fida.

Allow Office Scripts and Power Automate

Gli Office Scripts (script di automazione TypeScript) e i flussi di Power Automate possono leggere il contenuto della cartella di lavoro e portarlo fuori dall'applicazione. Lasciare questa opzione disattivata, a meno che il proprio flusso di lavoro non vi faccia affidamento.

Allow Microsoft connected experiences

Le esperienze connesse Microsoft (Microsoft connected experiences) inviano il contenuto delle celle ai servizi cloud di Microsoft. Questo riguarda Microsoft Copilot, Python in Excel, Smart Lookup (Ricerca intelligente), Editor, Translator (Traduttore) e Designer. Abilitarla solo se i clienti hanno bisogno di queste funzionalità e si accetta che il contenuto delle celle interessate possa essere elaborato nel cloud Microsoft.

Note

- Queste opzioni si applicano all'applicazione compilata, non al designer di XLS Padlock.
- Il blocco di queste superfici nasconde anche i relativi punti di accesso nella barra multifunzione di Excel (per esempio il menu a comparsa dei componenti aggiuntivi, la scheda Automate e il pulsante Copilot), in modo che agli utenti finali non venga proposto di usare una funzionalità disattivata.
- I componenti aggiuntivi COM e i componenti aggiuntivi Excel integrati sono gestiti separatamente. Vedere [Come gestire i componenti aggiuntivi di Excel](#).

Migliorare la protezione

Perché compilare in un EXE non è sufficiente

Quando si compila una cartella di lavoro in un'applicazione protetta, il file della cartella di lavoro originale (XLSX, XLSM, ecc.) viene cifrato e incorporato nel file .EXE risultante. Questo impedisce agli utenti di accedere direttamente al file originale.

La protezione di base non è sufficiente

Tuttavia, la semplice compilazione della cartella di lavoro fornisce soltanto un livello di sicurezza elementare. È come chiudere una porta senza girare la chiave. Per proteggere davvero la propria proprietà intellettuale, è necessario attivare le funzionalità di protezione avanzate offerte da XLS Padlock.

Per funzionare, XLS Padlock deve caricare la cartella di lavoro nella memoria di Excel. **Senza attivare i livelli di sicurezza aggiuntivi di XLS Padlock**, un malintenzionato esperto potrebbe potenzialmente estrarre la cartella di lavoro originale dalla memoria mentre l'applicazione è in esecuzione. Sebbene non si tratti di un'operazione banale, ciò evidenzia la necessità di una protezione più robusta.

Rendere inutilizzabile una cartella di lavoro estratta

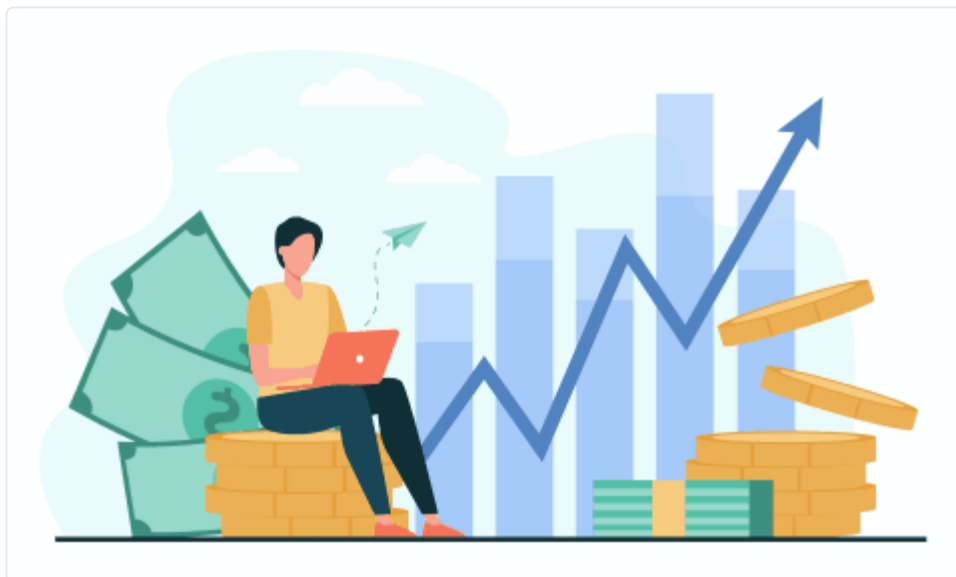
Quando si applicano correttamente le funzionalità di sicurezza di XLS Padlock, la cartella di lavoro estratta diventa inutilizzabile. Anche se qualcuno estrae il file, esso non funzionerà correttamente. La protezione è particolarmente forte per le cartelle di lavoro che contengono formule e/o codice VBA, poiché questi componenti critici vengono cifrati oppure compilati, impedendo di fatto qualsiasi uso non autorizzato o manomissione.

Le funzionalità di sicurezza più efficaci

Si consiglia di attivare le seguenti funzionalità per la protezione più forte:

- [Proteggere le formule con la protezione delle formule](#)
- [Proteggere il VBA con il VBA Compiler](#)
- [Prevenire i comuni attacchi VBA e OLE](#)
- [Vietare l'accesso all'editor VBA \(VBE\)](#)
- [Proteggere la cartella di lavoro con una password](#)
- [Disattivare i componenti aggiuntivi di Excel non necessari](#)

Controllo di accesso alla cartella di lavoro



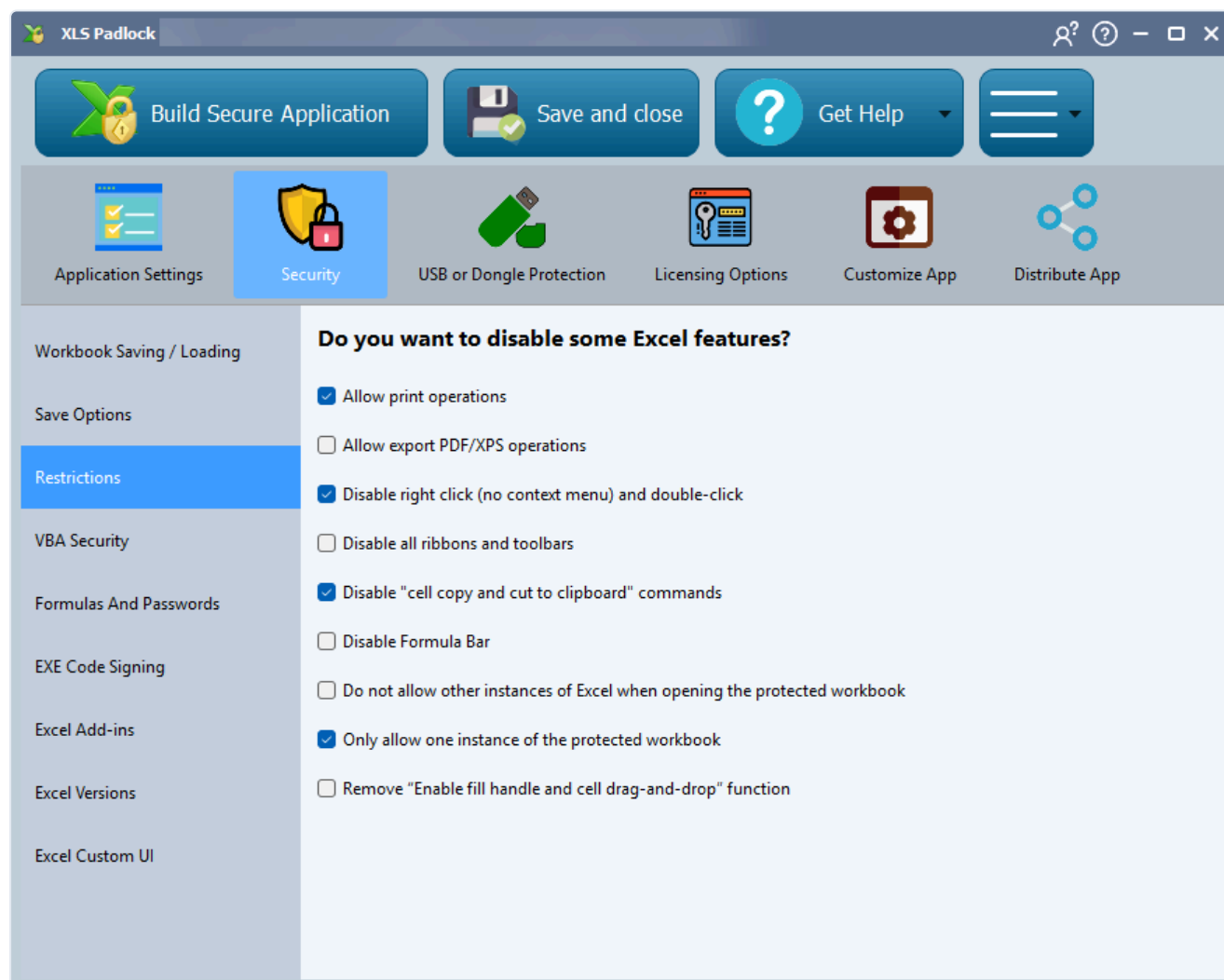
XLS Padlock offre numerose funzionalità di sicurezza per **limitare l'accesso alle vostre cartelle di lavoro Excel ai soli utenti autorizzati**. È possibile generare chiavi di attivazione, impostare versioni di prova a tempo limitato e molto altro. Un obiettivo primario di XLS Padlock è consentirvi di **vendere licenze per le vostre cartelle di lavoro**, fornendo funzionalità per ridurre la pirateria e la condivisione non autorizzata.

Questo argomento offre una panoramica delle funzionalità di sicurezza dedicate al controllo di accesso alle cartelle di lavoro:

- **Controllare chi può utilizzare la vostra cartella di lavoro:** [Configurare le chiavi di attivazione](#).
- **Impedire la condivisione bloccando una licenza su un singolo computer:** Utilizzare le [chiavi di attivazione con blocco hardware](#).
- **Gestire le attivazioni da remoto e disabilitare l'accesso:** Utilizzare l'[attivazione online](#), la [disattivazione](#) e la [convalida](#).
- **Creare versioni di prova della vostra cartella di lavoro:** Scoprire [come creare cartelle di lavoro di prova](#).
- **Limitare le azioni degli utenti:** [Vietare la stampa, l'esportazione, il clic destro e altro ancora](#).
- **Richiedere un dispositivo fisico per l'accesso:** Collegare l'applicazione a un [dongle](#) o a una [chiavetta USB](#).
- **Vendere licenze e abbonamenti online:** Integrare piattaforme e-commerce come [WooCommerce](#) o [FastSpring](#).

Restrizioni della cartella di lavoro Excel

Questo argomento illustra le opzioni disponibili nella sezione 'Security > Restrictions' di XLS Padlock, che consentono di limitare determinate azioni dell'utente nella cartella di lavoro protetta.



Allow Print Operations

Per impedire agli utenti di stampare la cartella di lavoro, disattivare l'opzione **Allow print operations** (Consenti operazioni di stampa) nelle impostazioni di [Security](#).

Se questa opzione è disattivata, qualsiasi tentativo di stampa verrà bloccato e l'utente vedrà un messaggio di errore "Printing is not allowed".

Allow Export to PDF/XPS

Per impedire agli utenti di esportare il contenuto della cartella di lavoro in file PDF o XPS, disattivare l'opzione "Allow export PDF/XPS operations" nelle impostazioni di [Security](#).

Disable Right-Click (and optionally Double-Click)

Per disattivare il menu contestuale del clic destro sulle celle della cartella di lavoro, attivare l'opzione "Disable right-click" nelle impostazioni di [Security](#).

Questo viene comunemente utilizzato per impedire agli utenti di copiare, incollare o modificare i dati e la formattazione delle celle.

Disable All Ribbons and Toolbars

L'opzione **Disable all ribbons and toolbars** (Disattiva tutte le barre multifunzione e le barre degli strumenti) nasconde tutte le schede integrate della barra multifunzione e la maggior parte dei comandi del menu Microsoft Office, creando così un'interfaccia semplificata, di tipo chiosco, per l'applicazione.

Questa funzionalità è compatibile con **Excel 2007 e versioni successive**. È possibile imporla impostando la [versione minima di Excel richiesta](#) per la cartella di lavoro.

Disable "Cell Copy and Cut to Clipboard" Commands

Attivare questa opzione per disattivare i comandi **Copy** e **Cut** per le celle. Questo impedisce agli utenti finali di copiare facilmente i dati dalla cartella di lavoro negli appunti per riutilizzarli in altre applicazioni.

Disable Formula Bar

L'opzione **Disable Formula Bar** (Disattiva la barra della formula) nasconde la barra della formula di Excel. Inoltre, l'opzione corrispondente nel menu Visualizza di Excel viene disattivata, in modo che gli utenti finali non possano riattivarla.

Per una sicurezza maggiore, si consiglia inoltre di [disattivare l'accesso all'editor VBA](#).

Do Not Allow Other Instances of Excel

Quando è attivata, questa opzione verifica se un'altra istanza di Excel è già in esecuzione nel momento in cui un utente avvia la cartella di lavoro compilata. Se viene rilevato un processo Excel esistente, un messaggio inviterà l'utente a chiuderlo prima di proseguire.

Questo garantisce che l'istanza di Excel che esegue la cartella di lavoro protetta sia la prima e l'unica, offrendo un ambiente più controllato.

Questa opzione si trova nelle impostazioni di [Security](#).

Only Allow One Instance of the Protected Workbook

L'opzione "Only allow one instance of the protected workbook", che si trova nelle impostazioni di [Security](#), impedisce agli utenti finali di eseguire più di un'istanza dell'applicazione protetta contemporaneamente. Se un utente tenta di aprire nuovamente l'applicazione mentre è già in esecuzione, la finestra esistente verrà portata in primo piano. Questa funzionalità impedisce efficacemente agli utenti di aprire più istanze della stessa cartella di lavoro protetta.

Nota di compatibilità

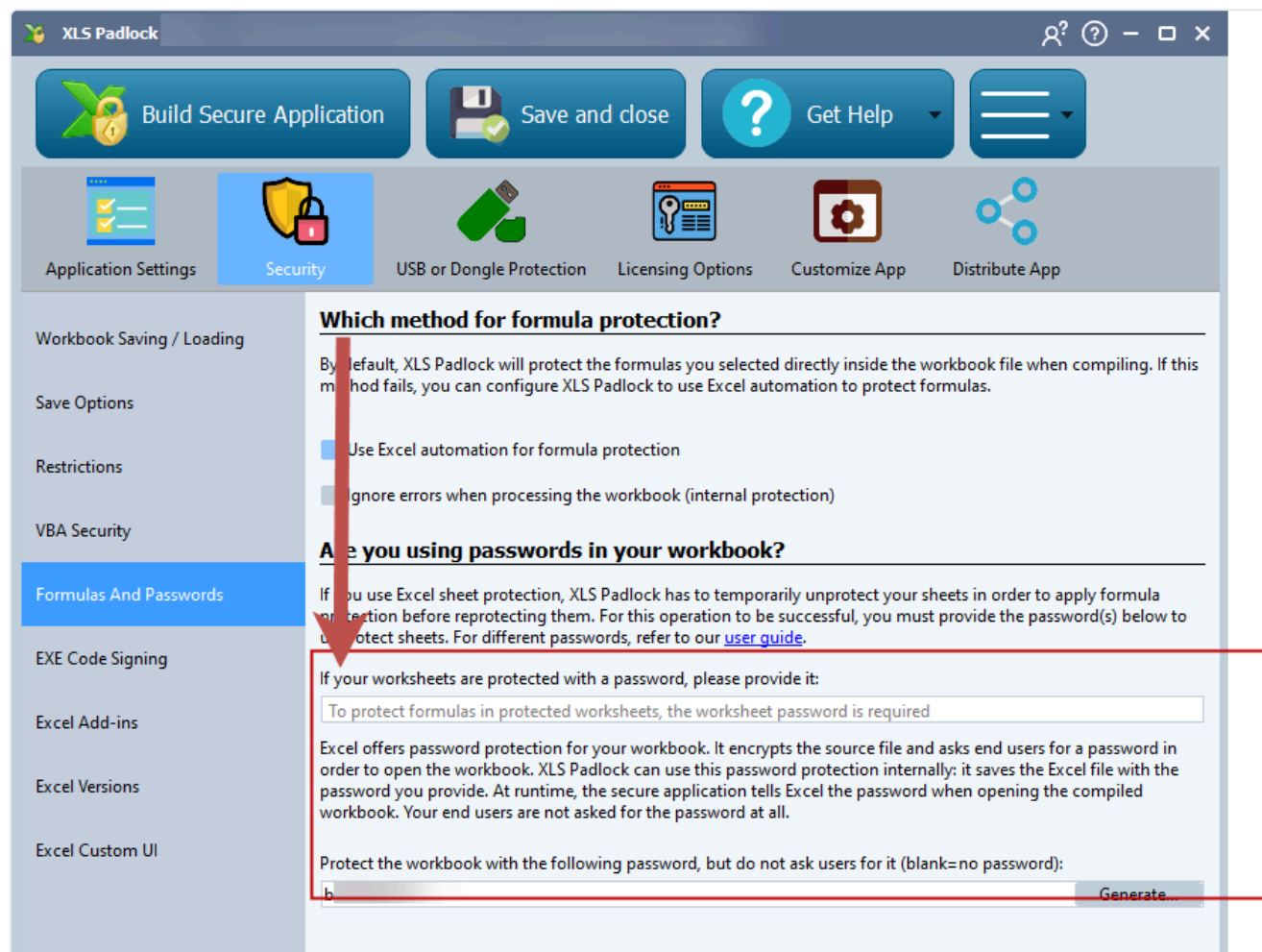
Nelle versioni di Excel precedenti alla 2013, verrà visualizzata una finestra di messaggio con la dicitura "An instance of this application is already running" anziché portare il focus sulla finestra esistente.

Remove "Enable fill handle and cell drag-and-drop" function

Questa opzione "Remove "Enable fill handle and cell drag-and-drop" function" impedisce agli utenti finali di utilizzare il quadratino di riempimento e il trascinamento delle celle nella cartella di lavoro protetta. È identica a quella presente nelle opzioni generali di Excel (Review", "Check accessibility", "Options accessibility", "Advanced", "Enable fill handle and cell drag-and-drop"), ma consente di prevalere sulla scelta dell'utente locale.

Protezione con password dei fogli

XLS Padlock consente di gestire sia le password della cartella di lavoro sia quelle dei fogli di calcolo per rafforzare la sicurezza e le funzionalità della sua applicazione protetta. Questa guida spiega come configurare queste impostazioni delle password.



The screenshot shows the XLS Padlock application window. The top navigation bar includes buttons for 'Build Secure Application', 'Save and close', 'Get Help', and a menu icon. Below this is a secondary navigation bar with icons for 'Application Settings', 'Security' (selected), 'USB or Dongle Protection', 'Licensing Options', 'Customize App', and 'Distribute App'. The left sidebar lists various settings categories, with 'Formulas And Passwords' highlighted. The main content area is titled 'Security' and contains two sections:

- Which method for formula protection?**

By default, XLS Padlock will protect the formulas you selected directly inside the workbook file when compiling. If this method fails, you can configure XLS Padlock to use Excel automation to protect formulas.

 - Use Excel automation for formula protection
 - Ignore errors when processing the workbook (internal protection)
- Are you using passwords in your workbook?**

If you use Excel sheet protection, XLS Padlock has to temporarily unprotect your sheets in order to apply formula protection before reprotecting them. For this operation to be successful, you must provide the password(s) below to unprotect sheets. For different passwords, refer to our [user guide](#).

If your worksheets are protected with a password, please provide it:

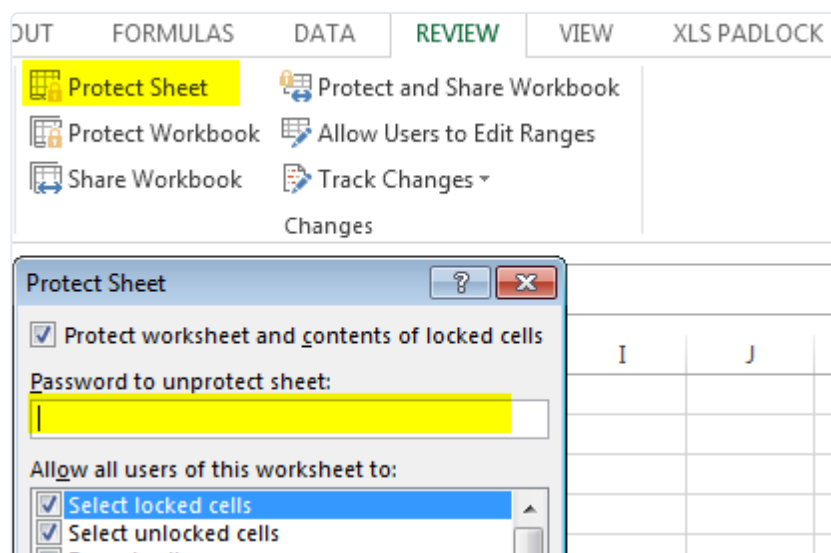
To protect formulas in protected worksheets, the worksheet password is required

Excel offers password protection for your workbook. It encrypts the source file and asks end users for a password in order to open the workbook. XLS Padlock can use this password protection internally: it saves the Excel file with the password you provide. At runtime, the secure application tells Excel the password when opening the compiled workbook. Your end users are not asked for the password at all.

Protect the workbook with the following password, but do not ask users for it (blank=no password):

Protezione con password dei fogli di calcolo

Se utilizza la protezione integrata dei fogli di Excel per limitare l'accesso alle celle, XLS Padlock deve conoscere la sua password per gestire i propri livelli di sicurezza.



Quando la [Protezione delle formule](#) di XLS Padlock è attiva, deve rimuovere temporaneamente la protezione dei fogli, applicare la propria protezione e poi proteggerli di nuovo. Per eseguire questa operazione con successo, deve fornire la password utilizzata per la protezione dei fogli in Excel.

NOTA

XLS Padlock riapplica automaticamente le stesse opzioni di protezione dei fogli definite nella cartella di lavoro originale.

Se utilizza un'unica password per tutti i fogli di calcolo protetti, la inserisca nel campo "If your worksheets are protected with a password, please provide it" (Se i fogli di calcolo sono protetti con una password, la fornisca).

Se utilizza password diverse per fogli diversi, deve fornirle utilizzando il seguente formato JSON:

```
{"worksheet 1 name": "Password1", "worksheet 2 name": "Password2", ...}
```

Per esempio:

```
{"Sheet1": "Hello World", "Sheet2": "Password2"}
```

Protezione silenziosa con password della cartella di lavoro

Excel offre una funzionalità che consente di cifrare l'intera cartella di lavoro con una password. XLS Padlock può gestire questa password al posto suo, offrendo un ulteriore livello di sicurezza senza interpellare gli utenti finali.

Quando imposta una password nel campo "Protect the workbook with the following password" (Proteggi la cartella di lavoro con la seguente password), XLS Padlock salva la cartella di lavoro con quella password. In fase di esecuzione, l'applicazione sicura fornisce automaticamente la password a Excel all'apertura della cartella di lavoro. **Agli utenti finali non viene mai richiesta la password.**

Questa combinazione aumenta la sicurezza della sua applicazione sovrapponendo la protezione di XLS Padlock alla cifratura nativa di Excel.

Può utilizzare il pulsante **Generate** (Genera) per creare una password complessa e casuale.

Non modificare la password dopo la distribuzione

Una volta distribuita un'applicazione protetta con una password specifica della cartella di lavoro, non la modifichi. In caso contrario, gli utenti finali non potranno aprire i dati salvati in precedenza, perché l'applicazione non sarà in grado di decifrarli.

Se lascia vuoto il campo della password, questa funzionalità sarà disattivata.

Combinare la protezione di Excel e di XLS Padlock

La protezione delle formule di XLS Padlock è pienamente compatibile con le funzionalità di protezione dei fogli integrate in Excel, come le celle bloccate o nascoste. Quando si utilizzano entrambe, le celle protette non mostreranno nulla nella barra della formula e le chiamate di funzione `PLEvalForm` sottostanti rimarranno nascoste.

	A	B	C	D
10		Durée jusqu'à échéance (en j)	90	t
11				
12				
13				
14		PRIX	$S \times N(d1) - (X/e^{rt}) \times N(d2)$	30.29 €
15				
16				
17		d1	$[\ln(S/X) + (r + 0,5 \sigma^2) \times t] / \sigma \sqrt{t}$	2.52

Avvertenza sulle dipendenze

Le formule protette che dipendono da altre celle non funzioneranno se tali celle vengono nascoste utilizzando l'attributo di formato cella "Hidden" di Excel. Evitare di utilizzare l'attributo nascosto sulle celle che costituiscono dipendenze delle formule protette.

Vedere anche:

- [Disattivare la protezione delle formule](#)

Protezione delle formule: Excel rispetto a XLS Padlock

Sebbene Excel consenta di nascondere le formule in un foglio protetto, questa protezione è limitata. XLS Padlock va molto oltre, sostituendo le formule con una funzione sicura disponibile solo quando l'applicazione è in esecuzione.

Nella cartella di lavoro compilata, le formule vengono sostituite nella barra della formula da funzioni anonime come `=PLEvalFormD(0;0)` o `=PLEvalFormD(1;COUNT($F7))`. I calcoli, tuttavia, continueranno a funzionare come previsto.

Prima della protezione, chiunque può vedere la formula:

The screenshot shows the Excel interface with the formula bar containing the following formula: `=LN(C6/C7)+(C9+0.5*C8^2)*(C10/360))/(C8*(C10/360)^(1/2))`. The spreadsheet below shows the following data:

	A	B	C	D
10		Durée jusqu'à échéance (en j)	90	t
11				
12				
13				
14		PRIX	$S \times N(d1) - (X/e^{rt}) \times N(d2)$	30.29 €
15				
16				
17		d1	$[\ln(S/X) + (r + 0,5 \sigma^2) \times t] / \sigma \sqrt{t}$	2.52
18				

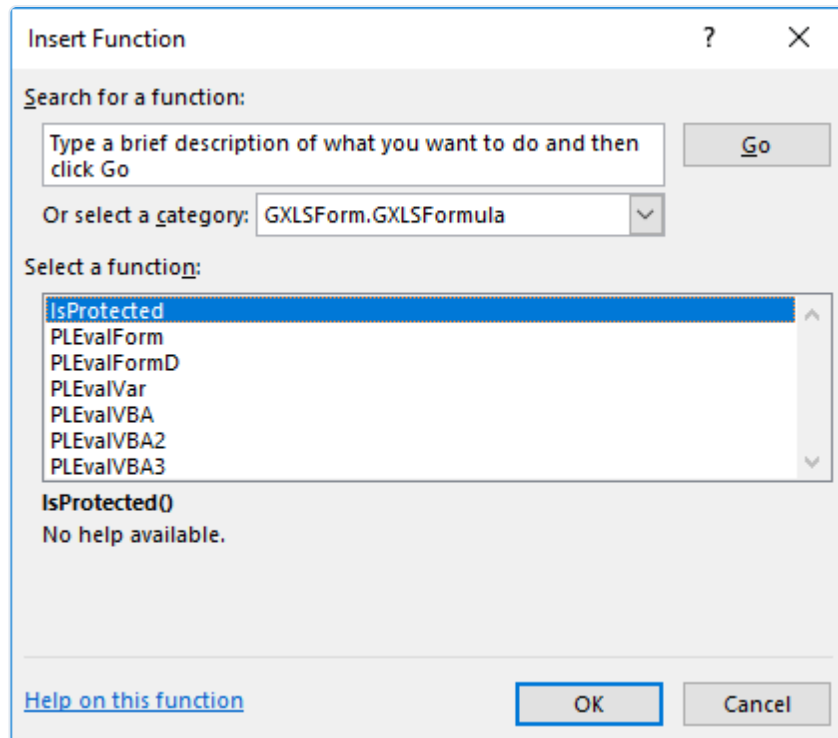
Dopo la protezione con XLS Padlock, la formula viene sostituita automaticamente pur restando funzionale:

The screenshot shows the Excel interface with the formula bar containing the secure function: `=XLSPadlockForm(0)`. The spreadsheet below shows the same data as the previous screenshot:

	A	B	C	D
10		Durée jusqu'à échéance (en j)	90	t
11				
12				
13				
14		PRIX	$S \times N(d1) - (X/e^{rt}) \times N(d2)$	30.29 €
15				
16				
17		d1	$[\ln(S/X) + (r + 0,5 \sigma^2) \times t] / \sigma \sqrt{t}$	2.52
18				

Questo aumenta notevolmente la sicurezza della cartella di lavoro. Poiché le formule originali vengono rimosse dal file della cartella di lavoro, le celle protette possono funzionare correttamente solo quando la cartella di lavoro viene aperta all'interno dell'applicazione sicura creata da XLS Padlock.

Anche in fase di progettazione, è possibile constatare che le funzioni di sicurezza di XLS Padlock come `PLEvalForm` e `PLEvalFormD` sono riconosciute da Excel:

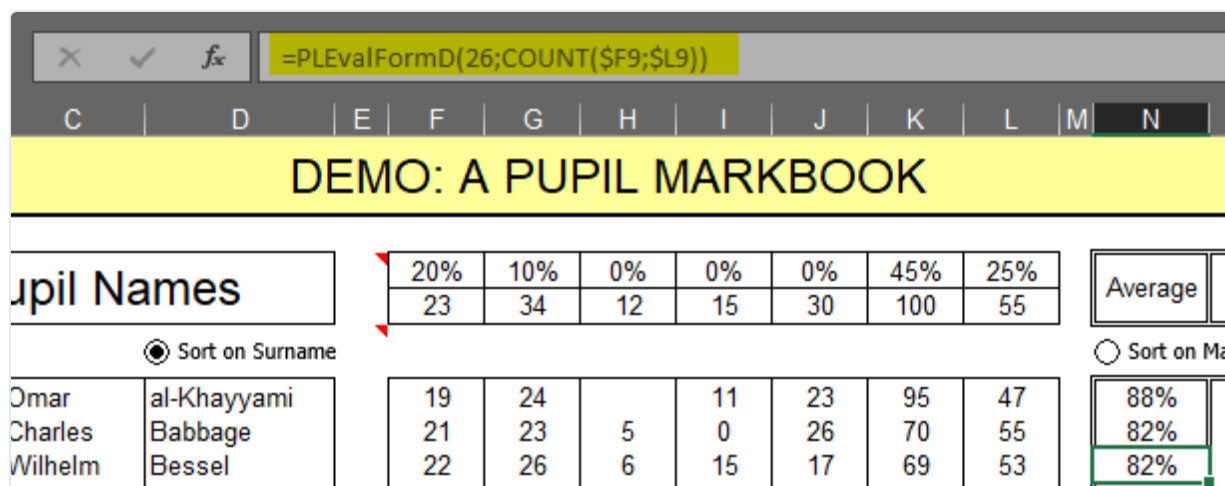


👉 Ulteriori informazioni sulla [protezione delle celle con XLS Padlock](#).

Proteggere le formule con XLS Padlock

Oltre alla protezione delle celle di Excel, **consigliamo vivamente di proteggere le formule importanti con la protezione delle formule propria di XLS Padlock.**

Dopo la protezione con XLS Padlock, le formule non appariranno più nella barra della formula, **pur rimanendo pienamente funzionali**. Al loro posto viene visualizzata una chiamata di funzione criptica come `PLEvalFormD` :

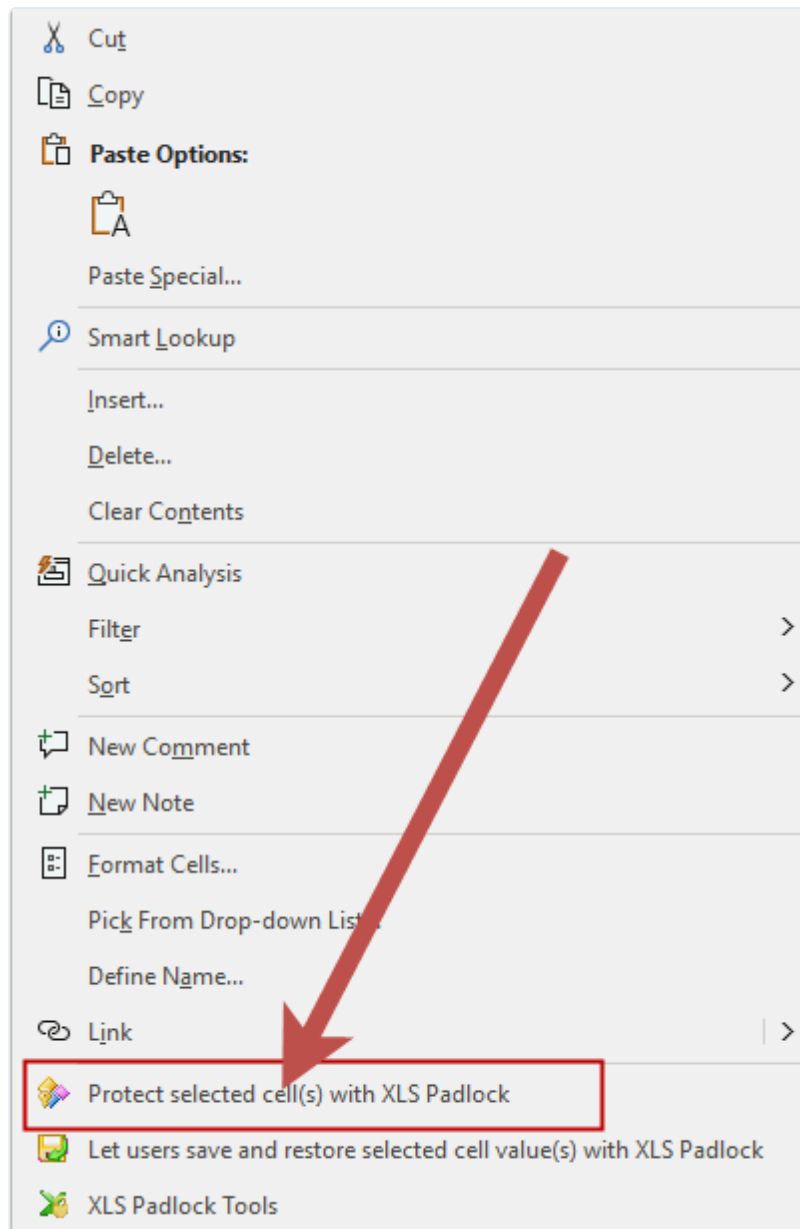


The screenshot shows an Excel spreadsheet titled "DEMO: A PUPIL MARKBOOK". The formula bar displays the protected formula `=PLEvalFormD(26;COUNT($F9;$L9))`. The spreadsheet contains a table of pupil names and their marks, with an "Average" column. The table is sorted by surname.

Pupil Names		20%	10%	0%	0%	0%	45%	25%	Average
Omar	al-Khayyami	19	24		11	23	95	47	88%
Charles	Babbage	21	23	5	0	26	70	55	82%
Wilhelm	Bessel	22	26	6	15	17	69	53	82%

👉 L'obiettivo è rendere la cartella di lavoro inutilizzabile se qualcuno riesce a [estrarre il file della cartella di lavoro dall'EXE compilato](#). Poiché le **formule protette da XLS Padlock non si trovano più nel file della cartella di lavoro**, la cartella di lavoro estratta non funzionerà correttamente.

XLS Padlock consente di decidere quali celle proteggere ed è possibile **selezionare e proteggere più celle contemporaneamente**.



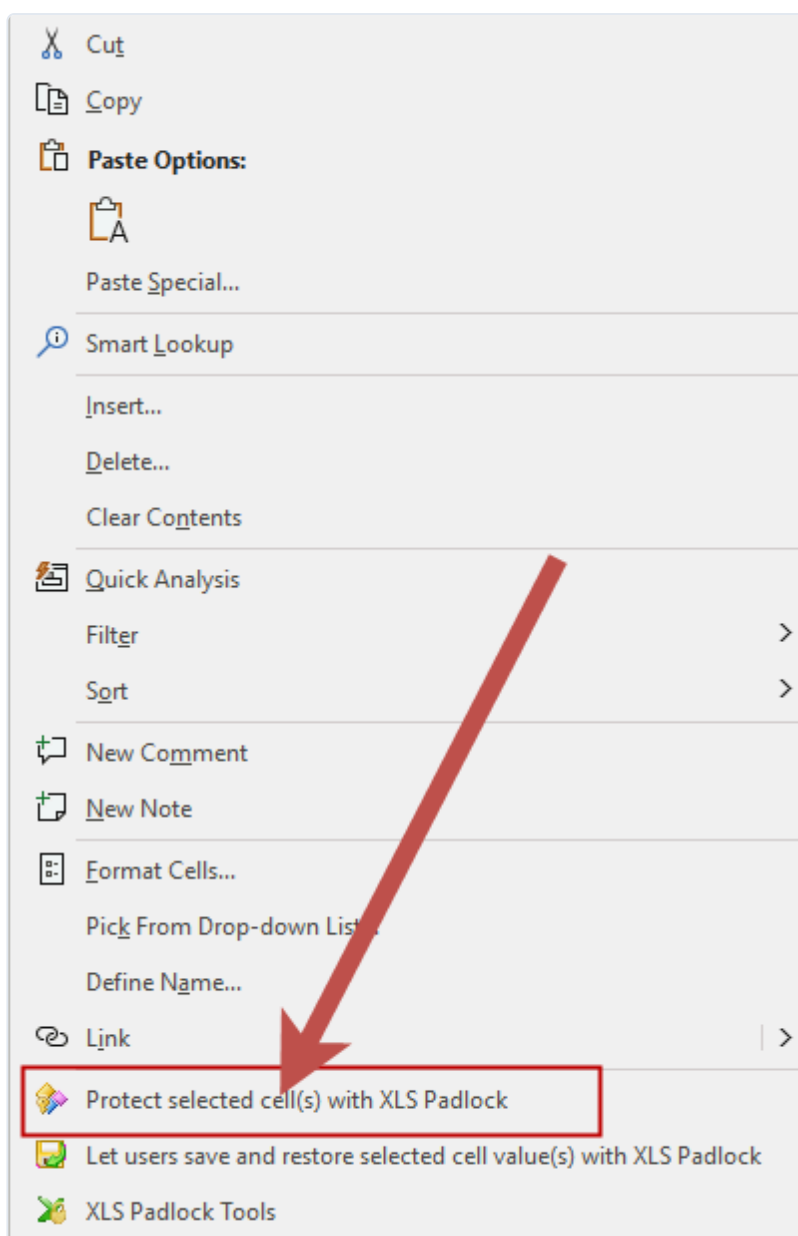
👉 Vedere come selezionare le celle da proteggere

Vedere anche:

- Scopri di più su come proteggere le formule in modo efficiente
- Disabilitare i componenti aggiuntivi Excel più comuni
- Utilizzare il VBA Compiler per una protezione ancora più forte

Contrassegnare le celle per la protezione delle formule

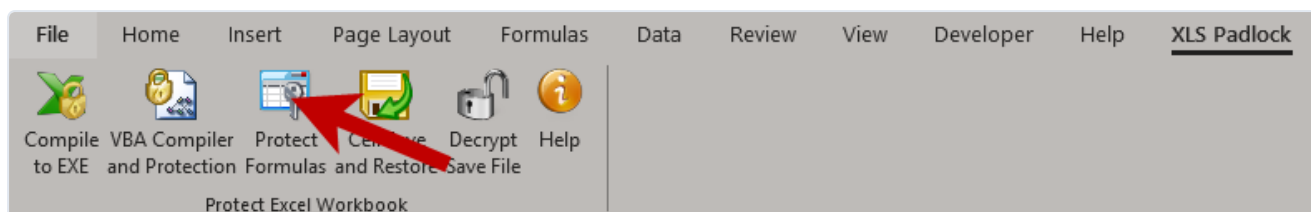
Per proteggere una formula, è sufficiente **fare clic con il pulsante destro su una o più celle** contenenti le formule che si desidera nascondere e selezionare **"Protect selected cell(s) with XLS Padlock"** (proteggi le celle selezionate con XLS Padlock) dal menu contestuale.



XLS Padlock confermerà quindi che le celle sono contrassegnate per la protezione.

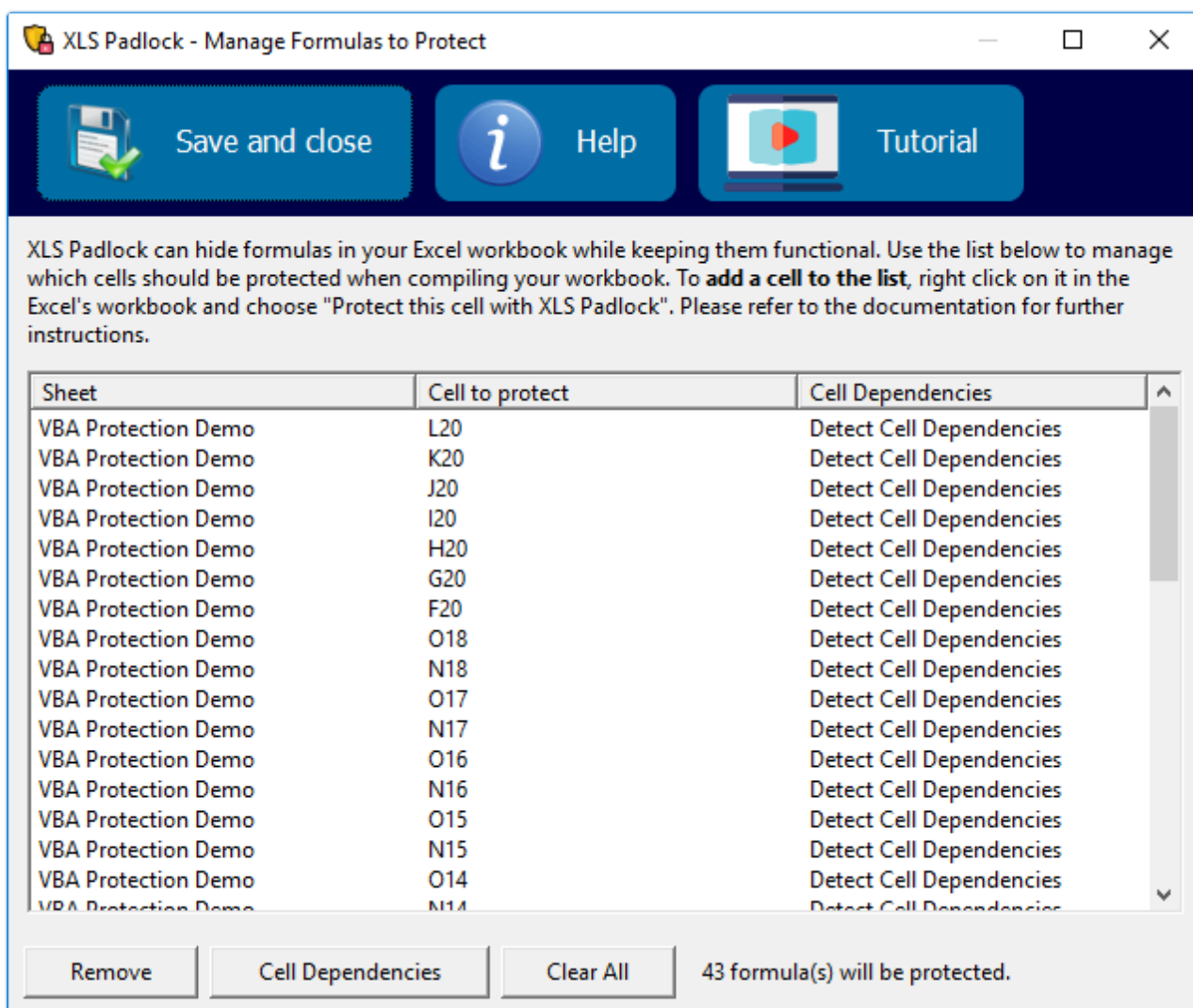
Panoramica delle celle protette

Per visualizzare una panoramica di tutte le celle protette, fare clic su **"Protect Formulas"** (proteggi formule) nella scheda o nel menu XLS Padlock:



Si apre così un elenco di tutte le celle configurate per la protezione. In questa finestra è possibile regolare il comportamento della protezione, rimuovere la protezione da celle specifiche o svuotare l'intero elenco.

Quando si compila la cartella di lavoro, XLS Padlock sostituisce tutte le formule elencate con chiamate di funzione generiche come `PLEvalForm(N)` e `PLEvalFormD(N, ...)`. **Le celle rimangono funzionali, ma gli utenti finali non possono scoprire le formule sottostanti.** Le formule originali non esistono più nella cartella di lavoro compilata; sono gestite dall'EXE stesso.



Dipendenze delle celle

Il pulsante **Cell Dependencies** (dipendenze delle celle) consente di controllare come viene applicata la protezione. Sono disponibili due scelte: "Detect Cell Dependencies" e "No".

Per impostazione predefinita, XLS Padlock rileverà tutti i riferimenti di cella e i nomi di intervallo presenti nelle formule (dipendenze delle celle) e genererà una funzione anonima che contiene tali riferimenti. Ciò consente a Excel di ricalcolare correttamente le celle protette. Ad esempio, se la formula da proteggere è `=A3^2`, XLS Padlock genererà una funzione come: `PLEvalFormD(1, COUNT(A3))`.

Se XLS Padlock non riesce a proteggere una cella, è possibile scegliere "**No**" per la relativa impostazione "Cell Dependencies". In questo caso verrà utilizzata una semplice funzione generica `PLEvalForm(N)`.

Limiti della protezione

Alcune formule complesse non sono supportate dalla protezione e non funzioneranno, visualizzando un `#ERROR!` o `#VALUE!` nella cella. Si raccomanda di testare a fondo le formule prima di distribuire la cartella di lavoro protetta. Prestare particolare attenzione ai seguenti limiti:

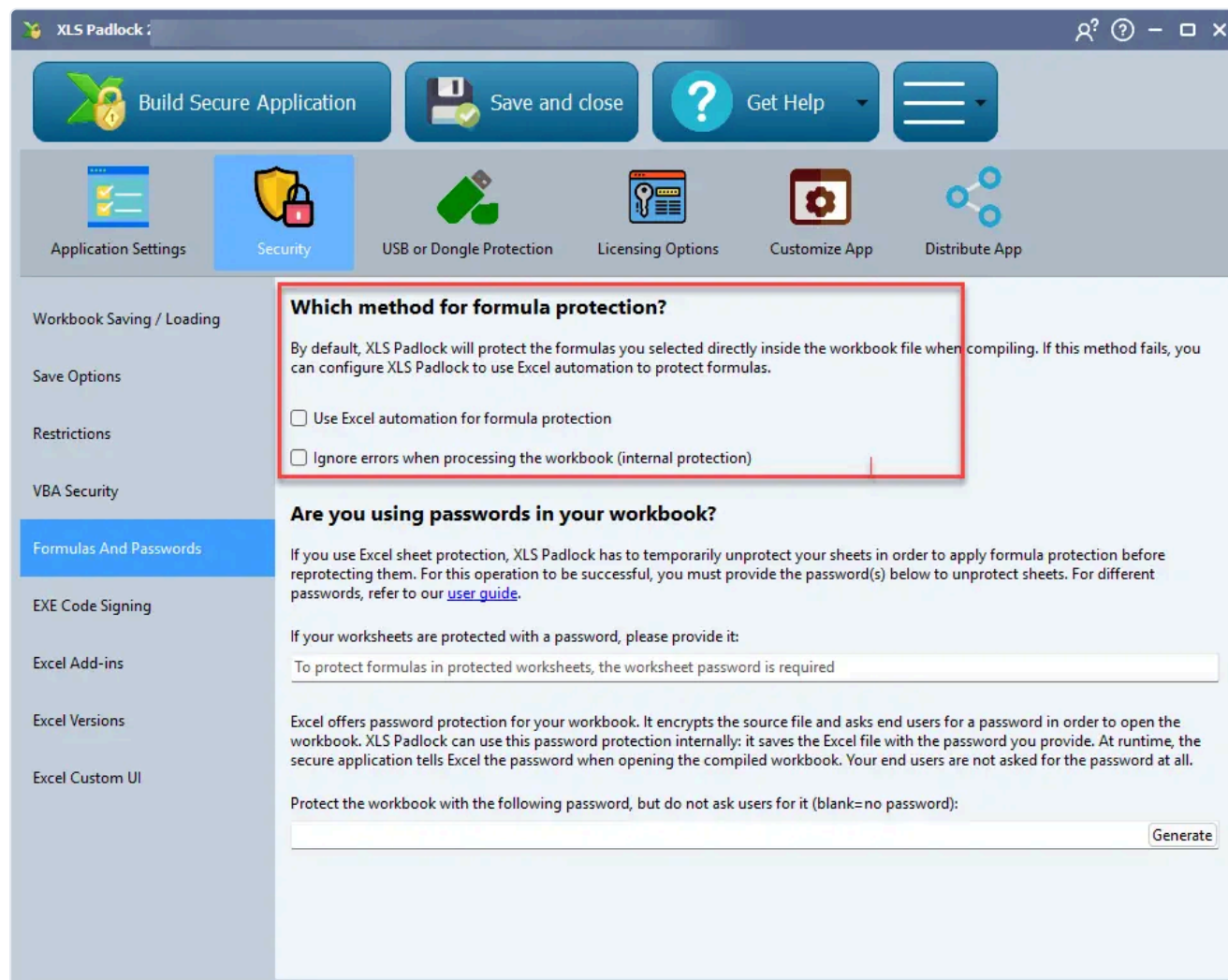
- La lunghezza della formula deve essere inferiore a 256 caratteri.
- La funzione `INDIRECT` non è supportata.
- Le formule non devono contenere alcuna funzione VBA definita dall'utente, ma solo normali funzioni Excel.
- La protezione delle formule deve essere usata con parsimonia. Proteggere solo le formule più importanti da cui dipende la cartella di lavoro. Non è consigliabile proteggere diverse migliaia di formule, poiché ciò aumenterà le dimensioni del file EXE e può rallentare Excel sui computer più datati.
- **La convalida dei dati delle celle e le regole di formattazione condizionale non sono supportate.**

👉 Vedere anche

- [Combinare la protezione del foglio Excel e la protezione XLS Padlock](#)

Metodo di protezione delle formule

Quando si attiva la protezione delle formule, XLS Padlock può utilizzare uno dei due metodi seguenti:



1. **Direct Modification (Recommended)** (modifica diretta, consigliata): questo metodo modifica le formule direttamente nel file sorgente della cartella di lavoro. È il metodo predefinito e funziona in modo affidabile nella maggior parte dei casi.
2. **Excel Automation** (automazione Excel): questo metodo utilizza l'automazione per controllare Excel e applicare la protezione.

Se il metodo consigliato non riesce, è possibile forzare XLS Padlock a utilizzare il metodo di automazione attivando "**Use Excel automation for formula protection**". Questo metodo viene utilizzato automaticamente anche per le cartelle di lavoro binarie (XLSB) e per i file di dimensioni molto grandi.

Come ultima risorsa, è possibile attivare "[Ignore errors when processing the workbook](#)", ma assicurarsi di testare a fondo l'applicazione compilata se si attiva questa opzione.

Disattivare la protezione delle formule

Poiché la protezione delle formule di XLS Padlock è incompatibile con Microsoft Excel 2000, è necessario attivare l'opzione **Disable formula and VBA protection** nella pagina Security se occorre creare applicazioni compatibili con questa versione.

Anche il VBA Compiler viene disattivato

L'attivazione di questa opzione disattiverà anche il compilatore di codice VBA.

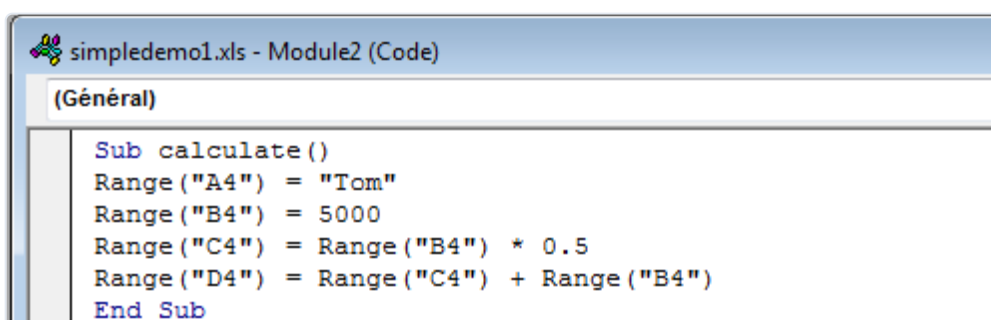
Informazioni sul VBA Compiler integrato

XLS Padlock dispone di un **compilatore VBA integrato** che trasforma gli script Basic in bytecode sicuro, rendendoli inaccessibili agli utenti finali. La compilazione di parti delle macro VBA rimuove il codice sorgente originale, rendendolo impossibile da copiare.

Tutorial video

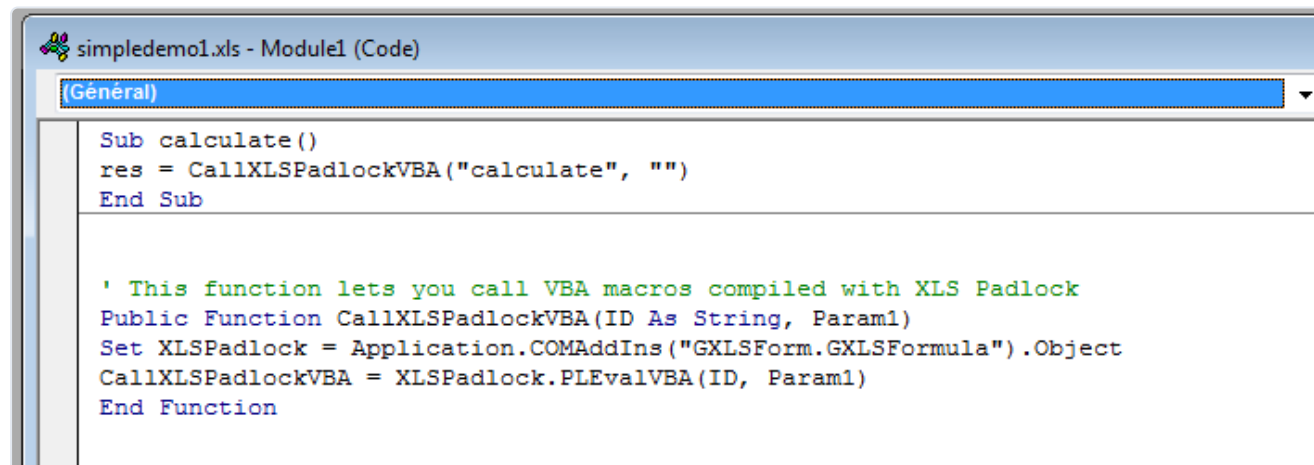
Per un tutorial sulla compilazione di una funzione VBA, è possibile [guardare questo video](#).

Ad esempio, si consideri questo codice originale:



```
simpledemo1.xls - Module2 (Code)
(Général)
Sub calculate()
Range("A4") = "Tom"
Range("B4") = 5000
Range("C4") = Range("B4") * 0.5
Range("D4") = Range("C4") + Range("B4")
End Sub
```

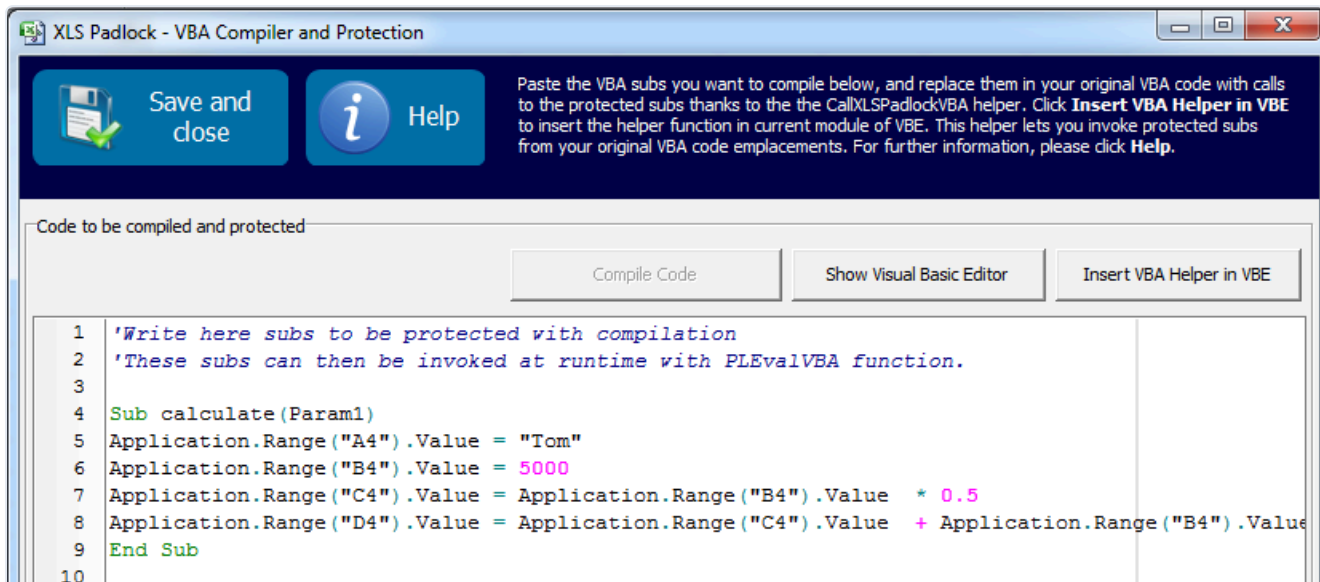
Una volta protetto, il sub `calculate()` originale viene sostituito da una chiamata a una funzione interna di XLS Padlock che esegue il bytecode compilato:



```
simpledemo1.xls - Module1 (Code)
(Général)
Sub calculate()
res = CallXLSPadlockVBA("calculate", "")
End Sub

' This function lets you call VBA macros compiled with XLS Padlock
Public Function CallXLSPadlockVBA(ID As String, Param1)
Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
CallXLSPadlockVBA = XLSPadlock.PLEvalVBA(ID, Param1)
End Function
```

Il codice originale viene spostato nel XLS Padlock VBA Editor e compilato da lì:



Si noti che potrebbero essere necessarie alcune modifiche, poiché il compilatore richiede l'uso esplicito dell'oggetto `Application` per [accedere agli oggetti Excel](#).

Il compilatore non è un semplice offuscatore; trasforma completamente il codice VBA in binario e lo memorizza in modo sicuro all'interno dell'applicazione. Combinato con la [protezione tramite password del progetto VBA](#), questo rende inutili gli strumenti di cracking delle password, poiché non possono accedere al file XLS originale.

Limitazioni del compilatore

- Potrebbe essere necessario modificare il codice VBA originale affinché possa essere compilato, poiché il compilatore VBA di XLS Padlock non è avanzato quanto l'interprete VB di Microsoft.
- Le macro intere generalmente non possono essere compilate, ma solo porzioni di esse.
- Alcuni oggetti e proprietà predefinite non sono accessibili.

👉 Vedere anche: [\[Scopri come scrivere e compilare codice VBA sicuro\]\(#chapter-writing-and-compiling-secure-vba-code\)](#)

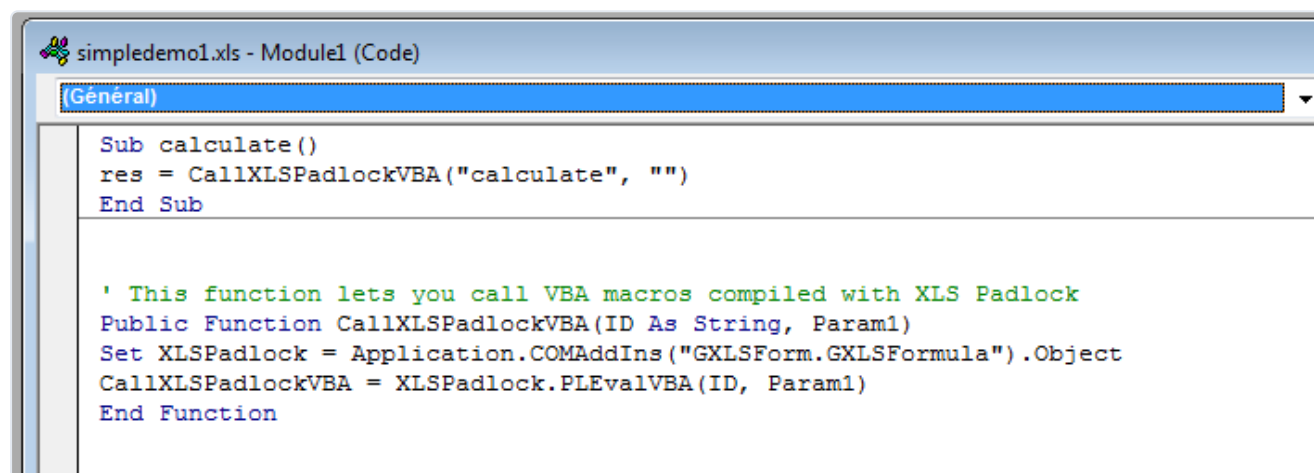
Protezione reale del codice VBA

Sebbene alcuni strumenti offrano l'offuscamento del VBA, si limitano a rendere il codice difficile da leggere. Non riescono a nascondere la logica fondamentale né a impedirne la copia.

👉 XLS Padlock va oltre, consentendo di **rimuovere completamente il codice VBA dalla cartella di lavoro, mantenendola al contempo funzionante**. Ciò è possibile grazie al VBA compiler integrato.

Il **VBA compiler di XLS Padlock converte il codice VBA in bytecode che può essere eseguito solo all'interno dell'applicazione protetta**. Se qualcuno riesce ad [accedere al file della cartella di lavoro](#), non vi troverà il codice originale, perché semplicemente non è presente. Al suo posto, il **bytecode compilato è memorizzato in modo sicuro all'interno dell'EXE**.

Ad esempio, la macro `calculate()` riportata di seguito conteneva in origine codice VBA proprietario. Dopo la compilazione, la macro presente nella cartella di lavoro contiene soltanto una chiamata al codice compilato. La logica originale è scomparsa.



```
simpledemo1.xls - Module1 (Code)
(Général)
Sub calculate()
res = CallXLSPadlockVBA("calculate", "")
End Sub

' This function lets you call VBA macros compiled with XLS Padlock
Public Function CallXLSPadlockVBA(ID As String, Param1)
Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
CallXLSPadlockVBA = XLSPadlock.PLEvalVBA(ID, Param1)
End Function
```

Ciò offre una **protezione robusta**, perché il codice VBA originale non esiste più nella cartella di lavoro. È stato sostituito da bytecode che non può essere copiato né utilizzato in un'altra cartella di lavoro Excel.

Il compromesso consiste nel fatto che è necessario **trasferire manualmente il codice VBA che si desidera proteggere nell'editor VBA di XLS Padlock**. La pratica consigliata è individuare le parti più critiche del codice, quelle che renderebbero la cartella di lavoro non funzionante se mancassero, e compilare soltanto tali sezioni.

👉 Vedere anche: [Scopri di più su come utilizzare il nostro VBA compiler](#)

Riferimento sintassi

Questo documento fornisce un riferimento per la sintassi VBA supportata dal [VBA compiler di XLS Padlock](#).

Struttura dello script

La struttura di uno script è costituita da dichiarazioni di funzioni e di routine (sub).

```
SUB DoSomething
    CallSomething
END SUB

FUNCTION MyFunction
    MyFunction = "Ok!"
END FUNCTION
```

Le istruzioni su una stessa riga possono essere separate dal carattere `:`.

Commenti

È possibile inserire commenti all'interno di uno script. Si può usare il carattere `'` oppure `REM`. Il commento si estende fino alla fine della riga.

```
' This is a comment before ShowMessage
ShowMessage("Ok")

REM This is another comment
ShowMessage("More ok!")
```

Identificatori

I nomi degli identificatori (variabili, funzioni, procedure, ecc.) devono iniziare con un carattere (a-z, A-Z) o con un trattino basso `_`. Possono essere seguiti da caratteri alfanumerici o da trattini bassi. I nomi non possono contenere altri caratteri né spazi.

Variabili

Non è necessario dichiarare le variabili in uno script, ma se lo si desidera è possibile dichiarare una variabile mediante la direttiva `DIM`.

```
SUB Msg(Param1)
  DIM S
  S = "Hello world!"
  ShowMessage(S)
END SUB
```

È inoltre possibile dichiarare variabili globali come `PRIVATE` o `PUBLIC`.

```
PRIVATE A
PUBLIC B
B = 0
A = B + 1
ShowMessage(A)
```

Le variabili dichiarate con `DIM` sono pubbliche per impostazione predefinita. Le variabili private non sono accessibili da altri script. Le variabili possono anche essere inizializzate al momento della dichiarazione:

```
DIM A = "Hello world"
DIM B As Integer = 5
```

Istruzioni di assegnazione

Le istruzioni di assegnazione utilizzano l'operatore `=` per assegnare un valore o il risultato di un'espressione a una variabile o a una proprietà di un oggetto.

```
MyVar = 2
Application.Range("C4").Value = "This " + "is ok."
```

Array

Il compilatore offre un supporto di base per i costruttori di array e per gli array variant. Per costruire un array, utilizzare i caratteri `[` e `]`. È possibile costruire un array a più indici annidando i costruttori di array.

Nel compilatore, gli array hanno indice a base 0.

```
NewArray = [ 2,4,6,8 ]
Num = NewArray[1] 'Num receives 4

MultiArray = [ ["green","red","blue"] , ["apple","orange","lemon"] ]
Str = MultiArray[0,2] 'Str receives "blue"
MultiArray[1,1] = "new orange"
```

Array dinamici:

```
' Create a dynamic array
DIM PTIM = VarArrayCreate([0,3000,0,5], 12)
' Assign a value:
PTIM[1,2] = 1530
```

Indici

Le stringhe, gli array e le proprietà di tipo array possono essere indicizzati mediante i caratteri [e]. Per esempio, se `Str` è una variabile di tipo stringa, l'espressione `Str[3]` restituisce il terzo carattere della stringa.

```
MyChar = MyStr[2]
MyStr[1] = "A"
MyArray[1,2] = 1530
```

Parole chiave e operatori

La sintassi Basic supporta:

- **Dichiarazioni:** `SUB...END SUB`, `FUNCTION...END FUNCTION`
- **Direttive:** `BYREF`, `DIM`
- **Condizionali:** `IF...THEN...ELSE...ELSEIF...END IF`, `SELECT CASE...END SELECT`
- **Cicli:** `FOR...TO...STEP...NEXT`, `DO...WHILE...LOOP`, `DO...LOOP...WHILE`, `DO...UNTIL...LOOP`, `DO...LOOP...UNTIL`
- **Operatori:** `^`, `*`, `/`, `AND`, `+`, `-`, `OR`, `<`, `>=`, `<=`, `=`, `>`, `<`, `DIV`, `MOD`, `XOR`, `SHL`, `SHR`
- **Gestione degli errori:** `TRY...EXCEPT`, `TRY...FINALLY`
- **Altro:** `EXIT`, costruttori di array `[1, 2, 3]`, accesso agli oggetti `ObjectName.Property`

Istruzioni If

Esistono due forme: `IF...THEN...END IF` e `IF...THEN...ELSE...END IF`. Se l'espressione è vera, vengono eseguite le istruzioni `THEN`. Se è falsa, vengono eseguite le istruzioni `ELSE` (se presenti).

```
IF J < 0 THEN
    Result = I/J
END IF

IF J = 0 THEN
    Exit
ELSE
    Result = I/J
END IF
```

Se l'istruzione si trova su una singola riga, non è necessario `END IF` :

```
IF J <> 0 THEN Result = I/J
```

Istruzioni `Select Case`

Se `selectorExpression` corrisponde a una delle espressioni `caseexpr` , vengono eseguite le rispettive istruzioni. In caso contrario, viene eseguita l'istruzione `CASE ELSE` .

```
SELECT CASE uppercase(Fruit)
  CASE "lime"
    ShowMessage("green")
  CASE "orange"
    ShowMessage("orange")
  CASE ELSE
    ShowMessage("black")
END SELECT
```

Istruzioni `For`

L'istruzione `FOR` ripete l'esecuzione di istruzioni finché un contatore non raggiunge un valore finale.

```
FOR counter = initialValue TO finalValue STEP stepValue
  Statements
NEXT
```

La parte `STEP` è facoltativa; se omessa, il valore del passo è 1.

```
FOR c = 1 TO 10 STEP 2
  a = a + c
NEXT
```

Istruzioni `While`

Un'istruzione `WHILE` ripete delle istruzioni finché una condizione di controllo è vera. La condizione viene valutata prima dell'esecuzione delle istruzioni.

```
WHILE (Data[I] <> X)
  I = I + 1
END WHILE
```

Istruzioni Do... Loop

Le istruzioni vengono eseguite **WHILE** (finché) un'espressione è vera, oppure **UNTIL** (fino a quando) un'espressione diventa vera. La condizione può essere verificata prima o dopo l'iterazione.

```
' Condition tested after
DO
  K = I mod J
  I = J
  J = K
LOOP UNTIL J = 0

' Condition tested before
DO WHILE I < 0
  ...
LOOP
```

Dichiarazioni di funzioni e di routine

Le dichiarazioni sono simili al Basic standard. Per restituire un valore da una funzione, assegnarlo alla variabile implicita avente lo stesso nome della funzione, oppure utilizzare l'istruzione **Return**. I parametri possono essere passati per riferimento mediante la direttiva **BYREF**.

```
SUB HelloWorld
  ShowMessage("Hello world!")
END SUB

FUNCTION Max(A,B)
  IF A > B THEN
    MAX = A
  ELSE
    MAX = B
  END IF
END FUNCTION

SUB SwapValues(BYREF A, B)
  DIM TEMP
  TEMP = A
  A = B
  B = TEMP
END SUB
```

Le routine (sub) e le funzioni sono pubbliche per impostazione predefinita, ma possono essere dichiarate come **PRIVATE**.

Ricettario API VBA

Questo argomento offre una raccolta completa di snippet di codice VBA ("ricette") per un controllo avanzato dell'applicazione protetta in fase di esecuzione. Utilizzando l'API VBA di XLS Padlock, è possibile recuperare informazioni di sistema, gestire il salvataggio e il caricamento, personalizzare l'interfaccia utente e interagire con le funzionalità online.

I due oggetti principali dell'API sono: `Application.COMAddIns("GXLSForm.GXLSFormula").Object` per la maggior parte delle informazioni in fase di esecuzione, e

`Application.COMAddIns("GXLS.GXLSPLock").Object` per le operazioni legate al salvataggio.

Ogni ricetta riportata di seguito fornisce una funzione o una subroutine VBA pronta all'uso.

Informazioni sull'applicazione e sul sistema

Queste ricette aiutano a recuperare informazioni sull'ambiente e sullo stato dell'applicazione.

Verificare se la cartella di lavoro è protetta

Da utilizzare per confermare che il codice sia in esecuzione all'interno di un'applicazione protetta creata da XLS Padlock.

```
Public Function XLSPadlockAvailable() As Boolean
    Dim XLSPadlock As Object
    On Error Resume Next
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    If Not XLSPadlock Is Nothing Then
        XLSPadlockAvailable = XLSPadlock.IsProtected()
    Else
        XLSPadlockAvailable = False
    End If
End Function
```

Ottenere il nome del file EXE dell'applicazione

Recupera il nome del file `.exe` in esecuzione.

```
Public Function GetEXEFilename() As String
    Dim XLSPadlock As Object
    On Error GoTo Err
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    GetEXEFilename = XLSPadlock.PLEvalVar("EXEFilename")
    Exit Function
Err:
    GetEXEFilename = ""
End Function
```

Ottenere la versione del file EXE e del prodotto

Recupera le stringhe di versione definite nella sezione "EXE Version Info" di XLS Padlock.

```
Public Function ReadVersionInfo(infoType As String) As String
    Dim XLSPadlock As Object
    On Error GoTo Err
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    ReadVersionInfo = XLSPadlock.PLEvalVar(infoType)
    Exit Function
Err:
    ReadVersionInfo = ""
End Function
```

Utilizzo: `ReadVersionInfo("ProductVersion")` restituisce la Product Version, e `ReadVersionInfo("FileVersion")` restituisce la File Version.

Ottenere il System ID dell'utente

Recupera il System ID univoco richiesto per le Activation Keys con blocco hardware (hardware-locked). Si tratta dell'identificatore che i clienti invieranno.

```
Public Function ReadSystemID() As String
    Dim XLSPadlock As Object
    On Error GoTo Err
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    ReadSystemID = XLSPadlock.PLEvalVar("SystemID")
    Exit Function
Err:
    ReadSystemID = ""
End Function
```

Ottenere i parametri della riga di comando

Recupera i parametri della riga di comando passati al file `.exe` dell'applicazione.

```
Public Function ReadParamStr(index As Integer) As String
    Dim XLSPadlock As Object
    On Error GoTo Err
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    ReadParamStr = XLSPadlock.PLEvalVar("ParamStr" & index)
    Exit Function
Err:
    ReadParamStr = ""
End Function
```

Utilizzo: `ReadParamStr(1)` restituisce il primo parametro, `ReadParamStr(2)` il secondo, e così via.

Salvataggio e caricamento delle cartelle di lavoro

Gestire i file di salvataggio sicuri (`.xlsc` o `.xlsce`) tramite programmazione.

Ottenere il percorso del file di salvataggio corrente

Recupera il percorso completo del file di salvataggio sicuro attualmente caricato.

```
Public Function GetSecureWorkbookFilename() As String
    Dim XLSPadlock As Object
    On Error GoTo Err
    Set XLSPadlock = Application.COMAddIns("GXLS.GXLSPLock").Object
    GetSecureWorkbookFilename = XLSPadlock.GetSaveFilename()
    Exit Function
Err:
    GetSecureWorkbookFilename = ""
End Function
```

Ottenere il percorso della cartella di salvataggio locale

Recupera il percorso della cartella locale in cui XLS Padlock memorizza le impostazioni dell'applicazione e i file di salvataggio sicuri.

```
Public Function GetStoragePath() As String
    Dim XLSPadlock As Object
    On Error GoTo Err
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    GetStoragePath = XLSPadlock.PLEvalVar("SPath")
    Exit Function
Err:
    GetStoragePath = ""
End Function
```

Eeguire codice VBA dopo il salvataggio

XLS Padlock può richiamare automaticamente la subroutine `XLSPadlock_OnAfterSave` dopo che un utente ha salvato il proprio lavoro. Ciò è utile per la registrazione su log, per mostrare un messaggio di conferma o per altre azioni successive al salvataggio.

```
Sub XLSPadlock_OnAfterSave(SaveFilename As String)
    MsgBox "The workbook has been successfully saved as: " & SaveFilename
End Sub
```

Aprire un file di salvataggio esistente con VBA

Aprire un file di salvataggio sicuro tramite programmazione. Si noti che questo caricherà il file in una nuova istanza di Excel.

```
Public Sub LoadXLSPadlockSaveFile(FilePath As String)
    Dim XLSPadlock As Object
    On Error Resume Next
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    If Not XLSPadlock Is Nothing Then
        XLSPadlock.PLOpenSaveFile FilePath
    End If
End Sub

' Example of a caller function that shows a file browser:
Sub Load_Old_Save()
    Dim strFileToOpen As String
    strFileToOpen = Application.GetOpenFilename( _
        Title:="Please choose a save file to open", _
        FileFilter:="Save Files (*.xlsc),*.xlsc")

    If strFileToOpen <> "False" Then
        LoadXLSPadlockSaveFile strFileToOpen
    End If
End Sub
```

Suggerire un nome di file per la finestra di dialogo di salvataggio

Imposta il nome di file predefinito che appare nella finestra di dialogo "Save As".

```
Public Sub SetSecureWorkbookFilename(Filename As String)
    Dim XLSPadlock As Object
    On Error Resume Next
    Set XLSPadlock = Application.COMAddIns("GXLS.GXLSPLock").Object
    If Not XLSPadlock Is Nothing Then
        XLSPadlock.SetDefaultSaveFilename Filename
    End If
End Sub

' Example: Set the default name to "my save.xlsx"
' Call SetSecureWorkbookFilename("D:\My Documents\my save.xlsx")
```

Salvare una copia sicura senza richiesta di conferma

Salvare una copia sicura della cartella di lavoro direttamente in un percorso di file specificato, senza mostrare la finestra di dialogo "Save As".

```
Public Function SaveSecureWorkbookToFile(Filename As String) As Boolean
    Dim XLSPadlock As Object
    On Error GoTo Err
    Set XLSPadlock = Application.COMAddIns("GXLS.GXLSPLock").Object
    SaveSecureWorkbookToFile = XLSPadlock.SaveWorkbook(Filename)
    Exit Function
Err:
    SaveSecureWorkbookToFile = False
End Function

' Example: Save the workbook directly
' If SaveSecureWorkbookToFile("D:\My Documents\my save.xlsx") Then
'     MsgBox "Workbook saved successfully."
' End If
```

Migrare i dati utente da una versione precedente

Se si pubblica una nuova versione della cartella di lavoro, potrebbe essere necessario importare i dati da un file di salvataggio creato con una versione precedente.

→ [Scopri come migrare i dati utente con VBA](#)

Interfaccia utente

Personalizzare gli elementi dell'interfaccia, come le finestre di dialogo.

Nascondere tramite programmazione le finestre di dialogo di attesa e caricamento

È possibile nascondere la finestra di dialogo "Loading workbook, please wait..." tramite una chiamata VBA, anziché attendere la sua chiusura automatica.

```
Public Sub HideLoadingDialog()  
    Dim XLSPadlock As Object  
    On Error Resume Next  
    Set XLSPadlock = Application.COMAddIns("GXLS.GXLSPLock").Object  
    If Not XLSPadlock Is Nothing Then  
        ' A call with Option "3" and Value "0" hides the dialog.  
        XLSPadlock.SetOption Option:="3", Value:="0"  
    End If  
End Sub
```

Attivazione e validazione online

Interagire con le funzionalità online di XLS Padlock.

Verificare la presenza di una connessione a Internet

Verifica se sul computer dell'utente è disponibile una connessione a Internet attiva.

```
Public Function IsInternetAvailable() As Boolean  
    Dim XLSPadlock As Object  
    On Error GoTo Err  
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object  
    IsInternetAvailable = (XLSPadlock.PLEvalVar("InternetConnected") = "1")  
    Exit Function  
Err:  
    IsInternetAvailable = False  
End Function
```

Ottenere il token di attivazione online

Recupera un hash del token di attivazione restituito dal server web dopo un'attivazione online riuscita.

```
Public Function GetValidationToken() As String  
    Dim XLSPadlock As Object  
    On Error GoTo Err  
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object  
    GetValidationToken = XLSPadlock.PLEvalVar("ValidationToken")  
    Exit Function  
Err:  
    GetValidationToken = ""  
End Function
```

Verificare se la validazione online è riuscita

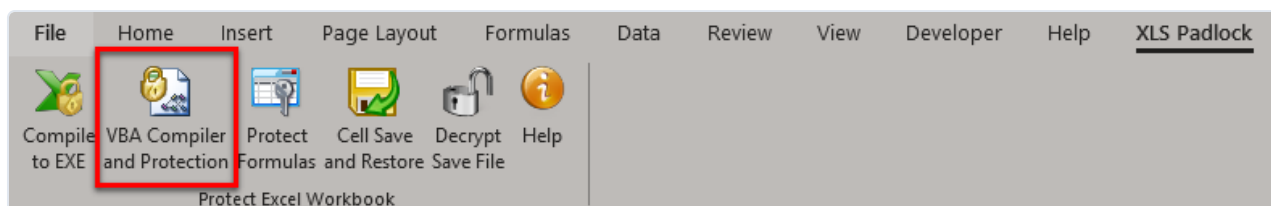
Ottiene il risultato dell'ultimo tentativo di validazione online.

```
Public Function IsValidationOK() As Boolean
    Dim XLSPadlock As Object
    On Error GoTo Err
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    IsValidationOK = (XLSPadlock.PLEvalVar("ValidationSuccess") = "1")
    Exit Function
Err:
    IsValidationOK = False
End Function
```

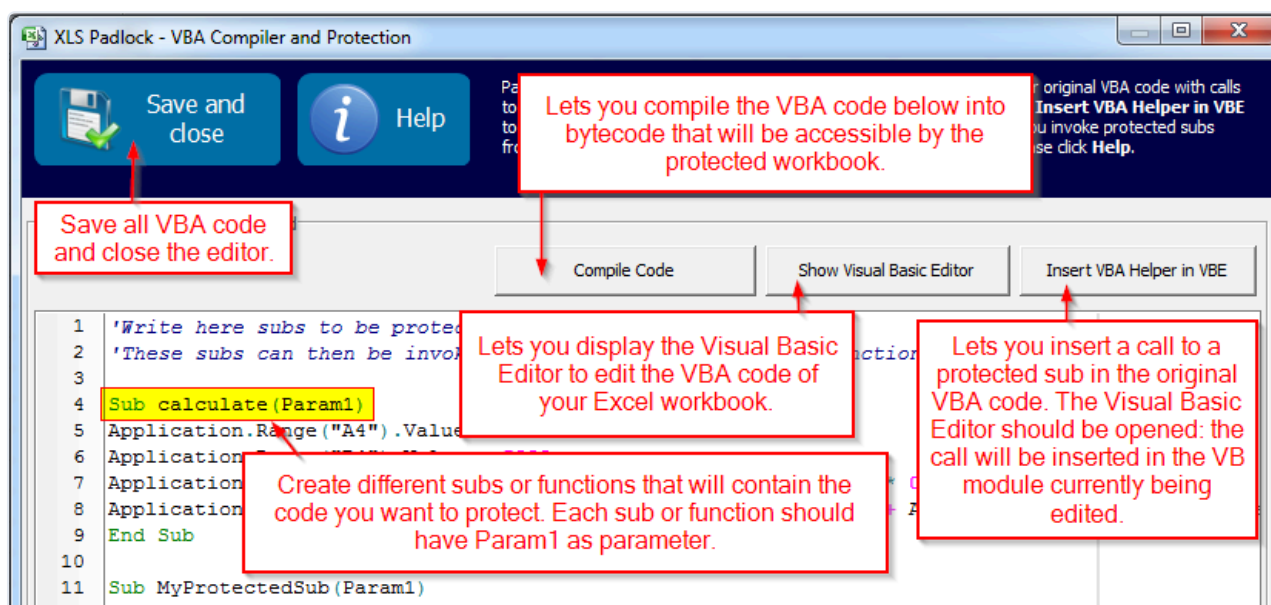
Scrivere e compilare VBA sicuro

Può **proteggere le parti sensibili del suo codice VBA compilandole in bytecode con XLS Padlock**. Consigliamo di combinare il bytecode compilato con il codice VBA esistente per un approccio equilibrato tra sicurezza e sviluppo.

1. In Excel, faccia clic su "VBA Compiler and Protection" per aprire l'editor e il compilatore VBA di XLS Padlock:



Si aprirà l'editor VBA di XLS Padlock:



2. Nell'area di testo, inserisca il codice VBA che desidera compilare. Può raggruppare il codice in `subs` e `functions`. Per semplificare questo processo, può aprire il Visual Basic Editor facendo clic su "Show Visual Basic Editor" e copiare/incollare il codice.

Per impostazione predefinita, tutti i subs e le functions hanno un parametro denominato `Param1`. È tuttavia possibile utilizzare più parametri.

3. Quando ha terminato, faccia clic su "**Compile Code**". Il codice VBA viene compilato istantaneamente in bytecode, che XLS Padlock conserverà fino a quando non compilerà il file EXE della sua applicazione.

SUGGERIMENTO

Se viene rilevato un errore, questo verrà visualizzato e il codice corrispondente sarà sottolineato in rosso.

1. Faccia clic su "Save and close" per chiudere l'editor.

👉 Dopo aver compilato il codice, dovrà **modificare le sue macro VBA originali affinché richi amino la versione compilata**. Scopri come procedere nella guida [Invocazione del codice VBA compilato in fase di esecuzione](#).

Richiamare il VBA compilato in fase di esecuzione

Per eseguire il [codice VBA compilato](#), è necessario richiamarlo dal codice VBA ordinario della cartella di lavoro tramite una funzione di supporto.

Per prima cosa, aprire il Visual Basic Editor e incollare la seguente funzione di supporto in un modulo della cartella di lavoro:

```
' This function lets you call VBA macros compiled with XLS Padlock
Public Function CallXLSPadlockVBA(ID As String, Param1 As Variant) As Variant
    Dim XLSPadlock As Object
    On Error Resume Next
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    CallXLSPadlockVBA = XLSPadlock.PLEvalVBA(ID, Param1)
End Function
```

Inserimento automatico

Questa funzione può essere inserita automaticamente nel modulo VB attivo facendo clic sul pulsante "Insert VBA Helper in VBE" (inserisci la funzione di supporto VBA nel VBE) nell'editor VBA di XLS Padlock.

La funzione di supporto `CallXLSPadlockVBA` accetta due parametri:

- **ID**: il nome del `Sub` o della `Function` compilata che si desidera invocare.
- **Param1**: un parametro opzionale che si desidera passare al codice compilato.

Esempio

Si immagini che il modulo VBA originale contenga il seguente codice:

```
Sub Calculate()
    Range("A4") = "Tom"
    Range("B4") = 5000
    Range("C4") = Range("B4") * 0.5
    Range("D4") = Range("C4") + Range("B4")
End Sub
```

Dopo aver spostato questo codice nel VBA Compiler, si sostituirebbe il sub originale con una chiamata alla funzione di supporto:

```
Sub Calculate()  
    Dim res As Variant  
    res = CallXLSpadlockVBA("Calculate", "")  
End Sub
```

In questo caso, si fornisce il nome del sub protetto, `Calculate`, e si passa una stringa vuota per il secondo parametro, poiché non viene utilizzato. La variabile `res` conterrà il risultato della chiamata, il che è utile soprattutto se il codice invocato è una `Function` che restituisce un valore.

Esecuzione in fase di progettazione

Una delle funzionalità più potenti di XLS Padlock è che questa configurazione funziona anche in fase di progettazione. Se si esegue il sub `Calculate()` dal VBE prima di compilare la cartella di lavoro, XLS Padlock eseguirà il codice protetto, consentendo di testare la propria logica senza dover prima compilare l'applicazione.

👉 Vedere anche: scoprire come [passare più di un parametro al codice compilato](#chapter-passing-more-parameters-to-the-compiled-vba-code).

Accedere agli oggetti Excel

Quando si scrive codice per il compilatore VBA, occorre utilizzare esplicitamente l'oggetto `Application` per accedere al modello a oggetti di Excel.

Per un riferimento dettagliato delle proprietà e dei metodi disponibili, consultare la [documentazione del Microsoft Office Dev Center](#).

Esempi

Riferimento a un Range

Per fare riferimento a una cella o a un intervallo di celle nel foglio di calcolo, occorre anteporre `Application` all'oggetto `Range`.

- **VBA standard:** `Range("A1:A4").Value = 2`
- **VBA compilato:** `Application.Range("A1:A4").Value = 2`

Utilizzo delle funzioni del foglio di calcolo

Per utilizzare le funzioni del foglio di calcolo di Excel, occorre utilizzare l'oggetto `Application.WorksheetFunction`.

```
Str1 = Application.WorksheetFunction.VLookup(Param1.ComboBox1, Application.Range("A2:C8"), 2, False)
```

Le funzioni richiedono tutti i parametri

Quando si chiamano funzioni integrate di Excel come `InputBox` dal codice compilato, **occorre specificare tutti i parametri**, poiché i parametri predefiniti non sono supportati.

Esempio completo: conversione di una macro

Ecco una macro VBA standard:

```
Sub test()  
    Dim qty As Integer  
    Dim price As Single, amount As Single  
    Range("A5").Value = "Item"  
    ActiveCell.Offset(1, 0).Select  
    ActiveCell.Value = InputBox("Enter the name of item")  
    ActiveCell.Offset(0, 1).Select  
    price = InputBox("Enter the price")  
    ActiveCell.Value = price  
    ' ... and so on  
End Sub
```

Ecco lo stesso codice modificato per il compilatore VBA di XLS Padlock. Si noti l'aggiunta dell'oggetto `Application` e dei parametri espliciti per `InputBox`.

```
Sub test(Param1)
    Dim qty As Integer
    Dim price As Single, amount As Single
    Application.Range("A5").Value = "Item"
    Application.ActiveCell.Offset(1, 0).Select
    Application.ActiveCell.Value = Application.InputBox("Enter the name of item", "Name", "")
    Application.ActiveCell.Offset(0, 1).Select
    price = Application.InputBox("Enter the price", "Price", "")
    Application.ActiveCell.Value = price
    ' ... and so on
End Sub
```

Il modulo VBA originale verrebbe quindi modificato per richiamare la versione compilata:

```
Sub test()
    res = CallXLSPadlockVBA("test", "")
End Sub
```

Passare array

Potete passare diversi tipi di variabili al codice VBA compilato, inclusi gli array statici.

Supponete di avere la seguente funzione nel VBA Compiler:

```
Function TestMultipleParams(Param1, Param2, Param3)
    MsgBox(Param2(1))
    TestMultipleParams = Param3 ^ 2
End Function
```

Nel vostro normale modulo VBA di Excel, potete chiamare questa funzione e passare un array.

Avvertenza

L'array deve essere definito come **Variant**.

```
Sub MySubSample4()
    Dim XLSPadlock As Object
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object

    Dim NomTableau(2) As Variant
    NomTableau(0) = "a"
    NomTableau(1) = "b"
    NomTableau(2) = "c"

    MsgBox XLSPadlock.PLEvalVBA3("TestMultipleParams", "Param1", NomTableau, 3)

    Set XLSPadlock = Nothing
End Sub
```

Passare più parametri

Per impostazione predefinita, la funzione di supporto utilizzata per eseguire il codice VBA compilato supporta un solo parametro.

```
Public Function CallXLSPadlockVBA(ID As String, Param1)
    Dim XLSPadlock As Object
    On Error Resume Next
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    CallXLSPadlockVBA = XLSPadlock.PLEvalVBA(ID, Param1)
End Function
```

Il metodo `PLEvalVBA` dell'oggetto `XLSPadlock` accetta due parametri: l'**ID** della sub/funzione compilata e un singolo parametro **Param1**.

Per passare più parametri, l'oggetto `XLSPadlock` fornisce metodi aggiuntivi:

- `XLSPadlock.PLEvalVBA2(ID, Param1, Param2)` per due parametri.
- `XLSPadlock.PLEvalVBA3(ID, Param1, Param2, Param3)` per tre parametri.

È inoltre possibile [passare più parametri utilizzando gli array](#).

Ad esempio, una funzione di supporto per passare due parametri avrebbe questo aspetto:

```
Public Function CallXLSPadlockVBA2(ID As String, Param1, Param2)
    Dim XLSPadlock As Object
    On Error Resume Next
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
    CallXLSPadlockVBA2 = XLSPadlock.PLEvalVBA2(ID, Param1, Param2)
End Function
```

Gestire gli errori

Durante l'esecuzione del codice VBA compilato possono verificarsi errori di runtime. Mentre il VBA standard utilizza l'istruzione `On Error` per la gestione degli errori, il VBA Compiler di XLS Padlock utilizza un blocco `Try...Except`.

Quando si verifica un errore all'interno del blocco `Try` (o in qualsiasi procedura da esso richiamata), il compilatore passa immediatamente al blocco `Except` per gestirlo.

Sintassi

```
Try
    ' ... Code to execute ...
Except
    ' ... Code to run if an exception occurs ...
End
```

SUGGERIMENTO

Un errore non gestito verrà mostrato dal VBA Compiler durante l'esecuzione, [a meno che non si disattivi questa opzione.](#)

Esempio di codice

```
NumberStr = ""
if InputQuery("Input", "Type an integer from 1 to 7", NumberStr) then
  try
    Number = StrToFloat(NumberStr)
  except
    raise("Not a valid number")
  end

  select case Number
    case 1
      ShowMessage("One")
    case 1 + 1
      ShowMessage("Two")
    case 4.5 / 1.5
      ShowMessage("Three")
    case 2 * 2
      ShowMessage("Four")
    case Length("xxxxx")
      ShowMessage("Five")
    case 3 + 3, 3 + 4
      ShowMessage("Six or Seven")
    case else
      ShowMessage("You did not type an integer from 1 to 7")
  end select
end if
```

OLE error 800A03EC

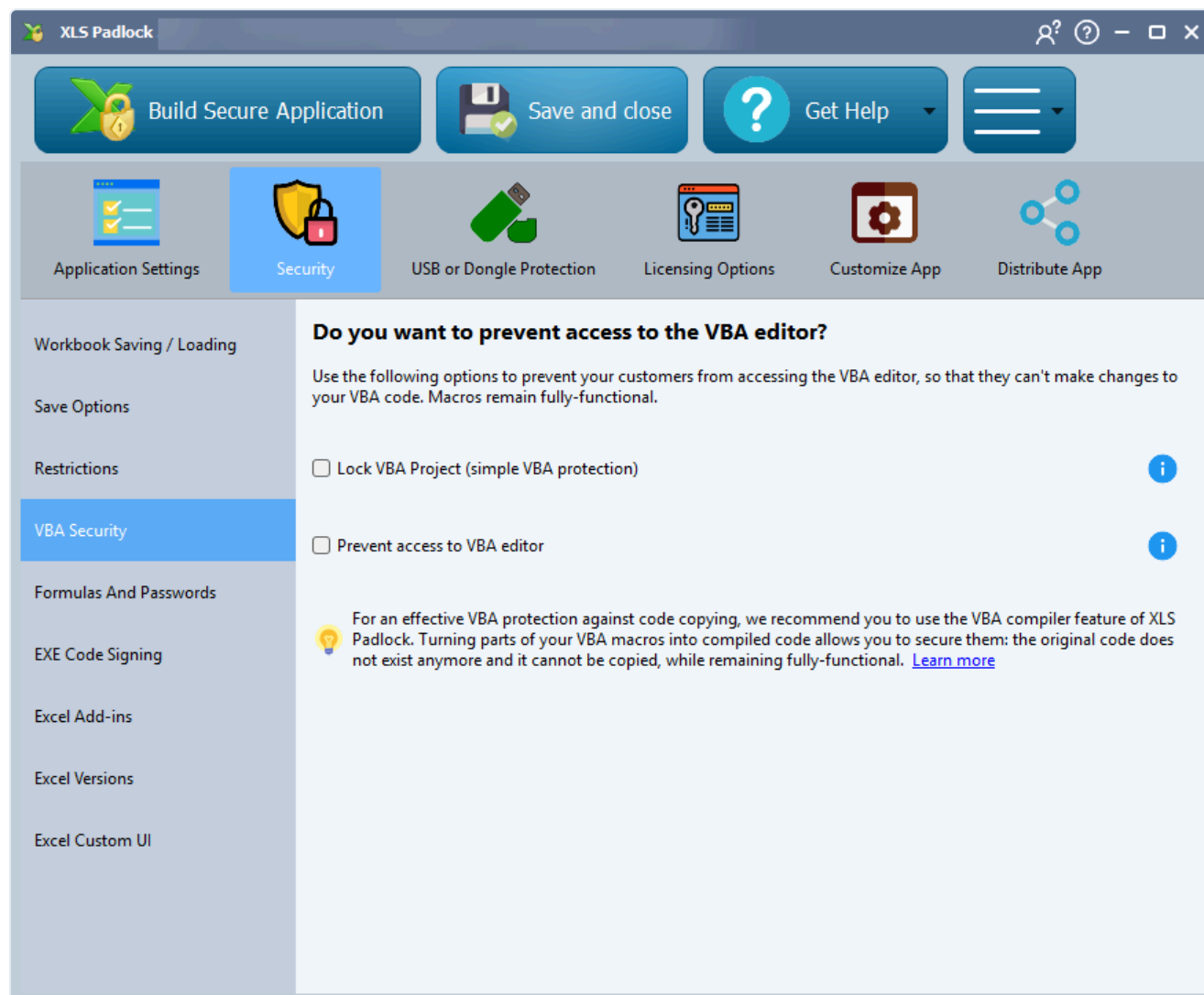
Per risolvere questo errore, assicurarsi che l'opzione « **Trust access to the VBA project object model** » sia attivata nelle impostazioni del Centro protezione (Trust Center) di Excel.

Per sapere come trovare questa opzione, consultare la pagina di assistenza ufficiale di Microsoft Office:

[Attivare o disattivare le macro nei file di Office](#)

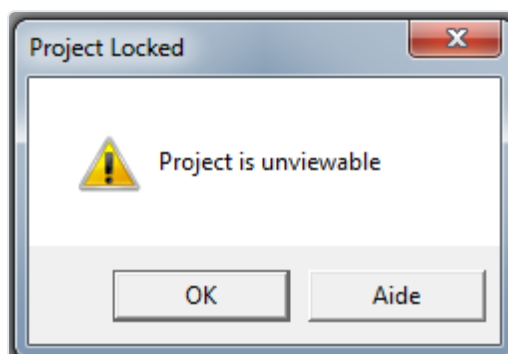
Protezione del codice VBA

Il codice VBA è una risorsa preziosa. XLS Padlock offre diversi livelli di protezione per metterlo al sicuro da manomissioni o furti. Di seguito sono riportate due opzioni di sicurezza fondamentali, seguite dalla nostra soluzione più avanzata per la massima protezione.



Opzione 1: bloccare il progetto VBA

Il passaggio più semplice consiste nel [bloccare il progetto VBA](#). Questo rende il codice non consultabile nell'editor Visual Basic. Se un utente tenta di accedere al progetto, incontrerà un messaggio di errore, bloccando di fatto l'ispezione occasionale del codice.



Opzione 2: impedire l'accesso all'editor VBA

Per un ulteriore livello di sicurezza, è possibile [impedire del tutto l'apertura dell'editor VBA](#). XLS Padlock monitora continuamente e chiude automaticamente qualsiasi tentativo di apertura del VBE, garantendo che anche gli utenti più esperti vengano bloccati.

Protezione definitiva: la compilazione VBA

Per il massimo livello di sicurezza, XLS Padlock consente di **compilare il codice VBA in codice binario nativo**. A differenza della semplice offuscazione, che rende soltanto il codice difficile da leggere, la compilazione trasforma la logica in un formato non leggibile da un essere umano, quasi impossibile da decodificare tramite reverse engineering. Il codice VBA originale viene rimosso dalla cartella di lavoro e sostituito da chiamate al codice compilato sicuro contenuto nell'EXE. Questa è la difesa definitiva per la proprietà intellettuale.

→ [Scopri di più sul nostro potente VBA Compiler](#)

Nascondere e bloccare il codice VBA

XLS Padlock offre una funzionalità [Lock VBA Project](#) dedicata. Tuttavia, se si sceglie di non utilizzarla, è comunque possibile ricorrere alla protezione nativa di Excel. XLS Padlock rende questo metodo più sicuro.

Come bloccare il progetto VBA in Excel

1. In Excel, aprire il Visual Basic Editor (VBE) premendo **Alt+F11** oppure selezionandolo dalla scheda **Developer**.
2. Nel VBE, accedere al menu **Tools** e selezionare **VBAProject Properties...**
3. Nella finestra di dialogo, accedere alla scheda **Protection**.
4. Selezionare la casella **Lock project for viewing** (blocca il progetto per la visualizzazione) e inserire una password robusta.

Una volta che la cartella di lavoro viene salvata e riaperta, gli utenti non potranno visualizzare né modificare il codice VBA senza la password.

Perché questo metodo è più sicuro con XLS Padlock

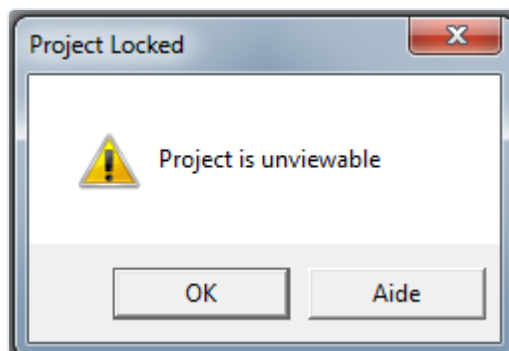
Normalmente, la protezione tramite password del VBA di Excel è considerata debole, poiché esistono numerosi strumenti in grado di rimuovere la password. Questi strumenti funzionano accedendo direttamente al file Excel.

Tuttavia, quando la cartella di lavoro viene compilata in un EXE con XLS Padlock, il file **.xls** o **.xlsm** originale viene cifrato e integrato all'interno dell'applicazione. Gli strumenti per la decifratura della password non possono accedere a questo file protetto, il che li rende inutili. Questa combinazione offre una difesa molto più solida per il codice VBA.

Bloccare il progetto VBA

L'opzione **Lock VBA Project** offre un modo semplice per impedire agli utenti finali di accedere al progetto VBA. Questa funzionalità contrassegna il progetto VBA come bloccato, rendendone impossibile la visualizzazione, l'accesso o la modifica. Non utilizza una protezione tramite password.

Se l'utente finale tenta di accedere a un progetto VBA bloccato, viene visualizzato il seguente messaggio di errore (Project Locked, Project is unviewable):



Rimuovere le password VBA esistenti

Se si utilizza questa opzione, si consiglia di rimuovere qualsiasi password esistente dal progetto VBA. Una password diventa ridondante quando il progetto è bloccato. Inoltre, l'uso di una password lunga con questa opzione può causare arresti anomali in Excel 2007.

Se si preferisce mantenere una password, valutare invece l'uso dell'opzione [Impedire l'accesso all'editor VBA](#chapter-forbid-access-to-the-vba-editor-vbe).

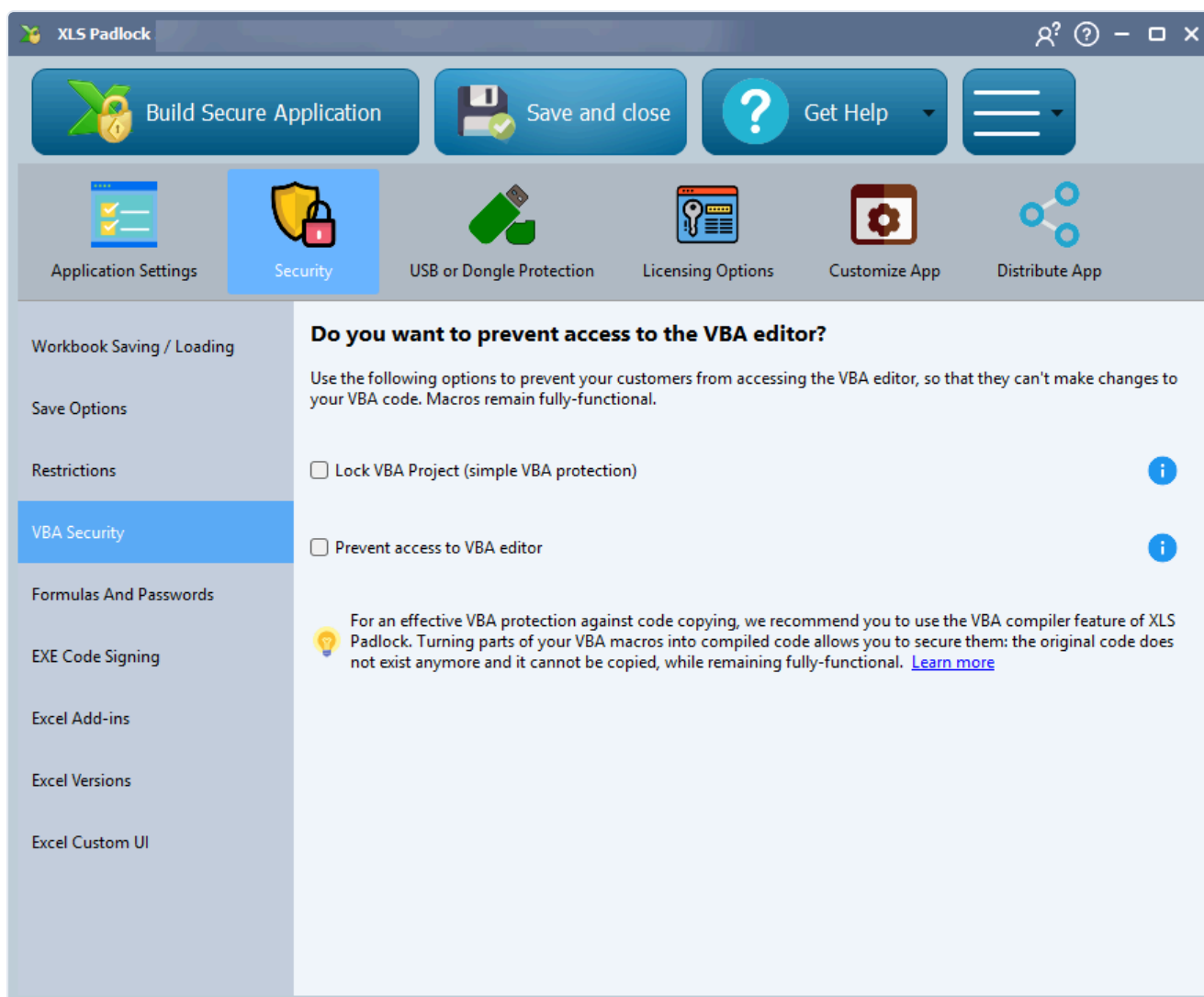
Notare che le macro restano funzionanti.

Questa opzione è compatibile con il nostro [compilatore VBA](#), che offre un livello di sicurezza più elevato per il codice VBA.

Vietare l'accesso al VBE

XLS Padlock offre opzioni per **impedire ai clienti di accedere all'editor VBA (VBE)**, bloccando di fatto la visualizzazione o la modifica del codice VBA, garantendo al contempo che le macro restino pienamente funzionanti.

Queste opzioni sono disponibili nella scheda "Security" di XLS Padlock.



Lock VBA Project (protezione VBA semplice)

Questa opzione blocca il progetto VBA in modo programmatico per ostacolarne l'accesso. Sebbene offra un livello di protezione di base, può essere aggirata da alcuni strumenti.

Prevent Access to VBA Editor

Si tratta di una misura di sicurezza più efficace. XLS Padlock esegue un controllo periodico e, se un utente tenta di aprire il Visual Basic Editor (VBE), XLS Padlock chiude immediatamente la finestra del VBE.

Questo impedisce persino il tentativo di visualizzare il codice.

Questa opzione è compatibile con l'opzione "Lock VBA Project" e con il nostro potente [compilatore VBA](#).

Per una protezione definitiva del codice VBA

👉 Per una protezione robusta contro il furto di codice e la retroingegneria, consigliamo vivamente di utilizzare la [funzionalità di compilatore VBA](#).

Compilando parti delle macro VBA, il codice sorgente originale viene rimosso e sostituito con codice binario. Questo codice compilato è estremamente difficile da retroingegnerizzare, garantendo la sicurezza dei propri algoritmi e della propria logica di business. È il modo più efficace per proteggere la proprietà intellettuale VBA.

Guida all'attivazione e alle licenze

Questa guida copre le varie opzioni relative a **chiavi di attivazione, licenza, convalida online e disattivazione** per le applicazioni protette.

The screenshot shows the XLS Padlock application window. The top bar contains buttons for 'Build Secure Application', 'Save and close', 'Get Help', and a menu icon. Below this is a navigation bar with icons for 'Application Settings', 'Security', 'USB or Dongle Protection', 'Licensing Options' (which is selected), 'Customize App', and 'Distribute App'. The main content area is titled 'Do you want to use activation keys?' and contains the following text:

You can require end users to activate the secure application with an activation key before they can access your protected workbook. Activation keys can have expiration date or limited number of runs, and they can also be **hardware-locked**: in that case, a key will only work on a given computer.

End users must enter an activation key in order to use the protected workbook

A hardware-locked key will only work on the computer of the user you have created it for. It won't work on any other machine, so it becomes useless if it is shared with others. The key is based on a **unique system ID** that depends on the user's computer hardware.

You can select the method to use to generate unique system IDs by clicking "**System ID Options...**" below.

Use hardware-locked keys (keys are based on a unique system ID)

Every protected workbook has a unique **Application Master Key**. This is what XLS Padlock uses to create keys associated to your workbook. It is strictly confidential: you should never give it to anyone. XLS Padlock automatically creates a new Master Key for each new project, but you can choose what you want.

Application Master Key:

To generate activation keys and set their properties (expiration date, system ID...), click **Key Generator** below.

At the bottom, there are three buttons: 'Key Generator', 'System ID Options...', and 'Clear Activation Data'.

Concetti fondamentali

Abilitazione delle chiavi di attivazione

Per richiedere agli utenti di disporre di una chiave per utilizzare l'applicazione, abilitare l'opzione: **"End users must enter an activation key in order to use the protected workbook"** (gli utenti finali devono inserire una chiave di attivazione per utilizzare la cartella di lavoro protetta). Questo è l'interruttore principale per attivare le funzionalità di licenza.

Application Master Key

Ogni cartella di lavoro protetta possiede un'**Application Master Key** univoca. XLS Padlock utilizza questa chiave per generare le chiavi di attivazione associate alla cartella di lavoro. È **strettamente riservata** e non deve mai essere condivisa. XLS Padlock crea automaticamente una nuova Master Key per ogni progetto.

Application GUID e Secret Key

- L'**Application GUID** è un identificatore univoco utilizzato dall'applicazione per memorizzare le proprie impostazioni e gestire i file di salvataggio sul computer di un utente.
- L'**Application Secret Key** viene utilizzata per cifrare e proteggere i file di salvataggio dell'utente (`.XLSC` o `.XLSCE`). Garantisce che i file di salvataggio creati con l'applicazione possano essere aperti solo dall'applicazione stessa.

IMPORTANTE

Se si modifica l'Application GUID o la Secret Key, i file di salvataggio precedenti non saranno più accessibili dagli utenti.

Pubblicazione di un aggiornamento che deve invalidare i vecchi salvataggi degli utenti

Per forzare una nuova versione a ignorare qualsiasi file di salvataggio creato con le versioni precedenti, ad esempio quando la struttura della cartella di lavoro è cambiata a tal punto che i vecchi salvataggi non devono essere ricaricati, generare una **nuova Application Secret Key** prima di ricompilare, ma **lasciare invariato l'Application GUID** affinché le chiavi di attivazione esistenti rimangano valide. Al primo avvio da parte dell'utente, il vecchio file di salvataggio verrà rilevato come incompatibile e all'utente verrà chiesto di caricare invece la cartella di lavoro originale incorporata. Consultare [Aggiornamenti della cartella di lavoro](#) per il flusso di lavoro completo di aggiornamento.

SUGGERIMENTO

Lo stato di attivazione (prova / registrato) può essere recuperato tramite codice VBA.

Consentire all'utente finale di cambiare la chiave di attivazione

Un utente può cambiare la propria chiave di attivazione eseguendo l'applicazione con il [parametro da riga di comando](#) `-enterkey` (ad esempio, `MYAPP.EXE -enterkey`). Questo è utile per sostituire una chiave che sta per scadere.

Cancellazione dei dati di attivazione

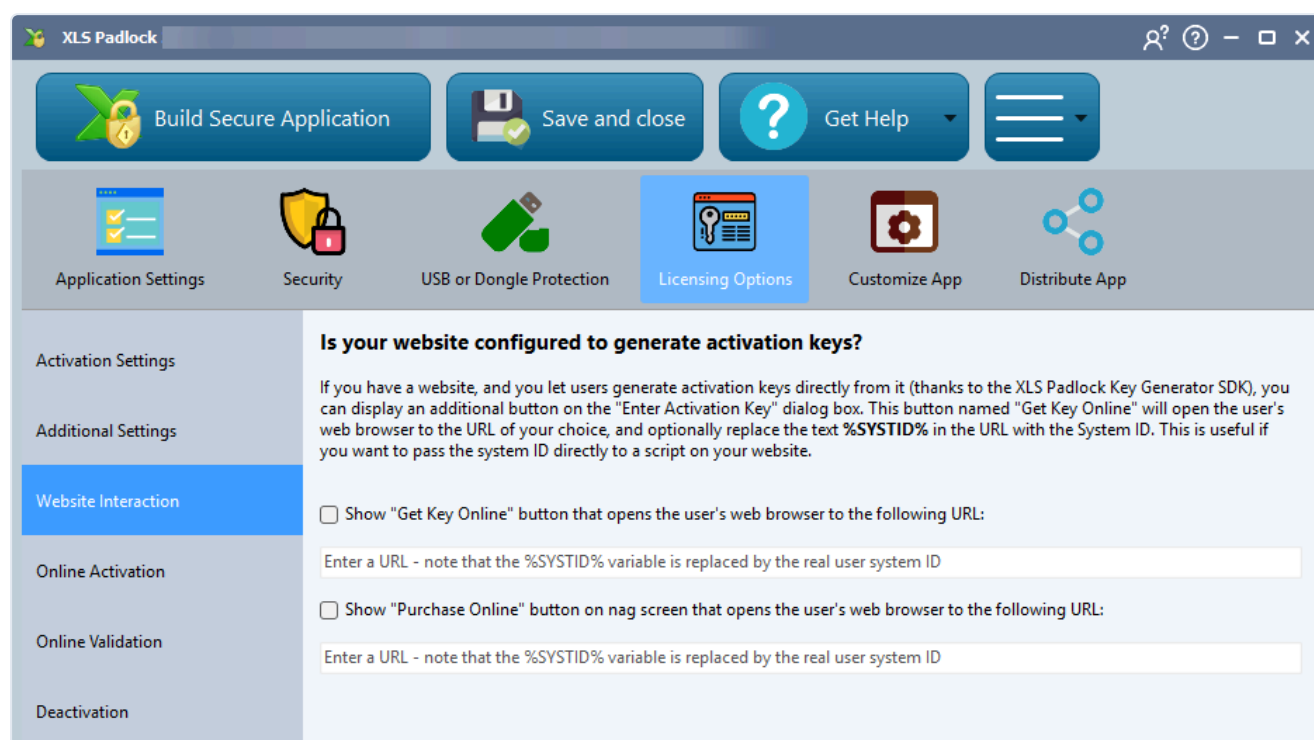
A scopo di test sulla propria macchina, il pulsante "**Clear Activation Data**" (cancella i dati di attivazione) nel Key Generator reimposta tutte le informazioni di attivazione memorizzate localmente.

Chiavi bloccate sull'hardware

Una chiave bloccata sull'hardware funzionerà **solo sul computer specifico** per il quale è stata generata, rendendola inutile se condivisa. La chiave si basa su un **System ID univoco** derivato dall'hardware del computer dell'utente.

- **Opzioni del System ID:** è possibile selezionare quali componenti hardware (CPU, indirizzo MAC, ecc.) vengono utilizzati per generare il System ID. Mantenere la coerenza di queste opzioni dopo aver distribuito l'applicazione.
- **Ottenere il System ID:** quando un utente esegue l'applicazione, gli verrà mostrato il proprio System ID, che dovrà inviare a voi. A quel punto si utilizza questo ID nel Key Generator per creare la sua chiave.
- **Automazione online:** per evitare lo scambio manuale dei System ID, è possibile utilizzare le funzionalità di [attivazione online](#).

Interazione con il sito web



Pulsante Get Key Online

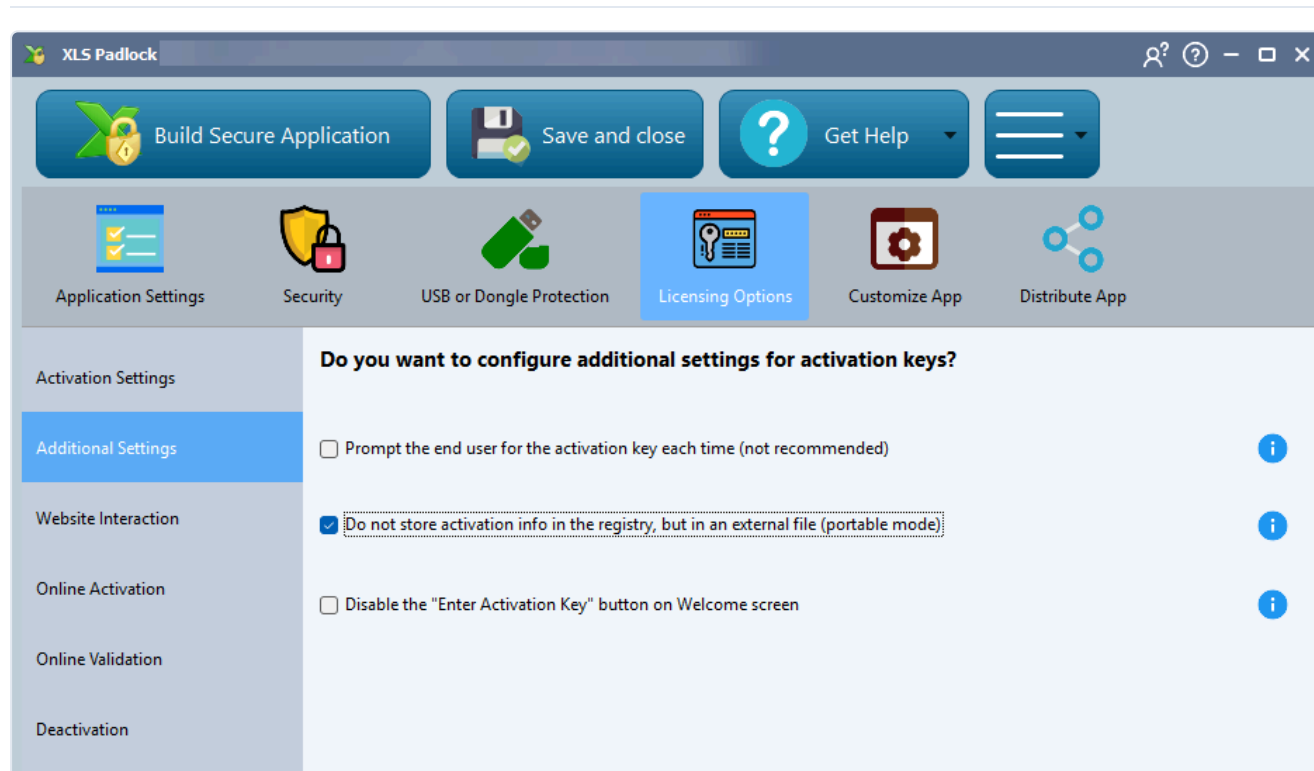
È possibile visualizzare un pulsante "Get Key Online" (ottieni chiave online) nella finestra di attivazione. Questo pulsante apre il browser web dell'utente a un URL specificato dall'utente. È possibile utilizzare il segnaposto `%SYSTID%` nell'URL per trasmettere automaticamente il System ID dell'utente allo script di generazione delle chiavi del proprio sito web.

URL di esempio: `https://www.yourwebsite.com/getkey.php?systid=%SYSTID%`

Mostrare il pulsante "Purchase Online"

Per le versioni di prova con una schermata di promemoria (nag screen), è possibile aggiungere un pulsante "Purchase Online" (acquista online) che indirizza l'utente verso il proprio negozio o la propria pagina di acquisto.

Impostazioni aggiuntive



Modalità portatile

Abilitare l'opzione **"Do not store activation info in the registry, but in an external file (portable mode)"** (non memorizzare le informazioni di attivazione nel registro, ma in un file esterno, modalità portatile). Questo crea un file `.lic` nascosto nella stessa cartella dell'EXE, consentendo di eseguire l'applicazione da un'unità USB senza scrivere nel registro di Windows.

IMPORTANTE

Questa modalità non è consigliata per le versioni di prova, poiché l'eliminazione del file `.lic` reimposta il periodo di prova.

Richiedere la chiave di attivazione all'utente finale ogni volta

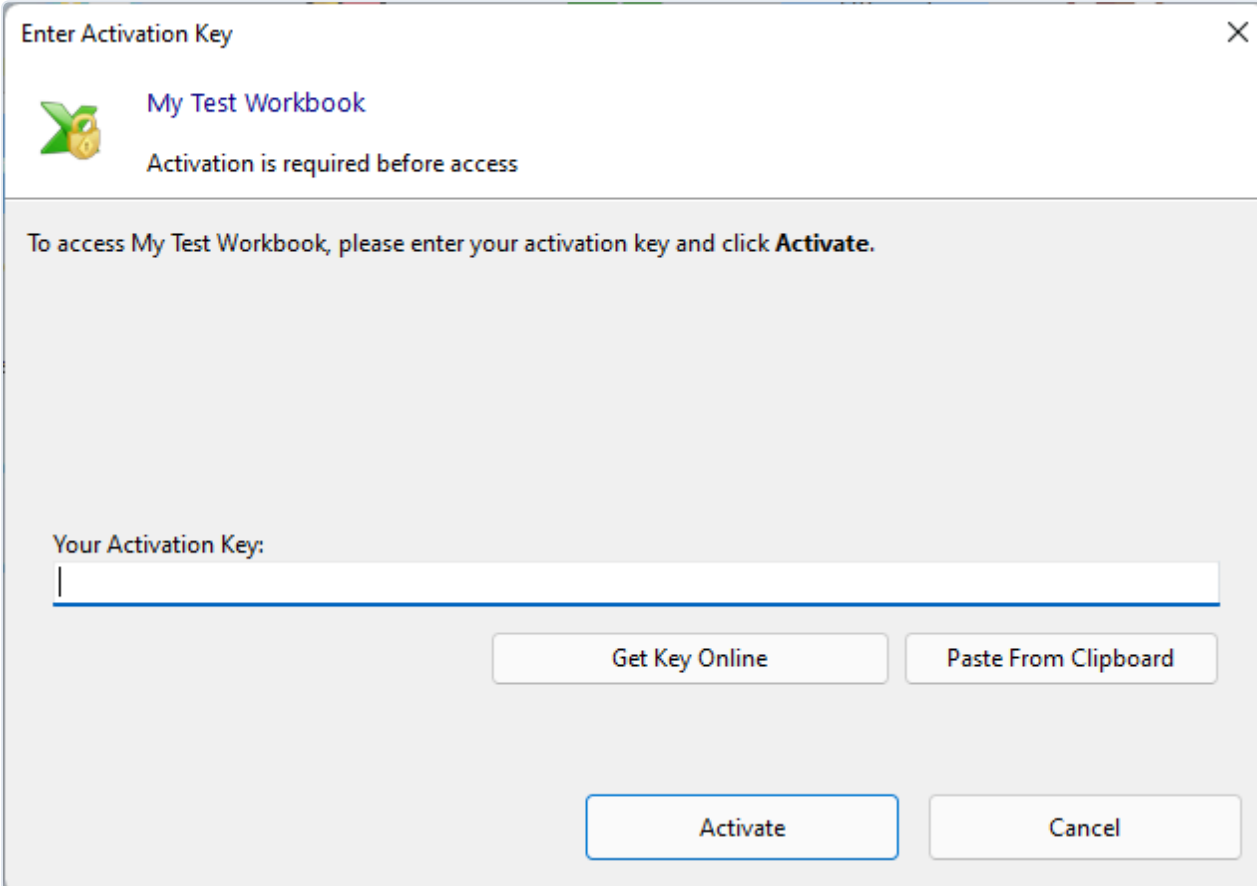
Abilitando "Prompt the end user for the activation key each time" (richiedi la chiave di attivazione all'utente finale ogni volta), gli utenti dovranno inserire la chiave di attivazione ogni volta che aprono l'applicazione. Non è consigliato, perché è fastidioso per il cliente e non è possibile impostare funzionalità di scadenza per la chiave.

Disabilitare il pulsante "Enter Activation Key" nella schermata di benvenuto


Questo rimuoverà la possibilità per gli utenti finali di inserire una chiave di attivazione quando aprono l'applicazione della cartella di lavoro Excel. Tuttavia, non è consigliato utilizzare questa opzione.

Chiavi di attivazione

Le chiavi di attivazione sono una funzionalità centrale di XLS Padlock e offrono un **potente sistema di licenza per le tue cartelle di lavoro Excel**. Questo sistema consente di richiedere agli utenti finali di attivare l'applicazione con una chiave prima di poter accedere alla cartella di lavoro protetta:



Enter Activation Key

 My Test Workbook
Activation is required before access

To access My Test Workbook, please enter your activation key and click **Activate**.

Your Activation Key:

Get Key Online Paste From Clipboard

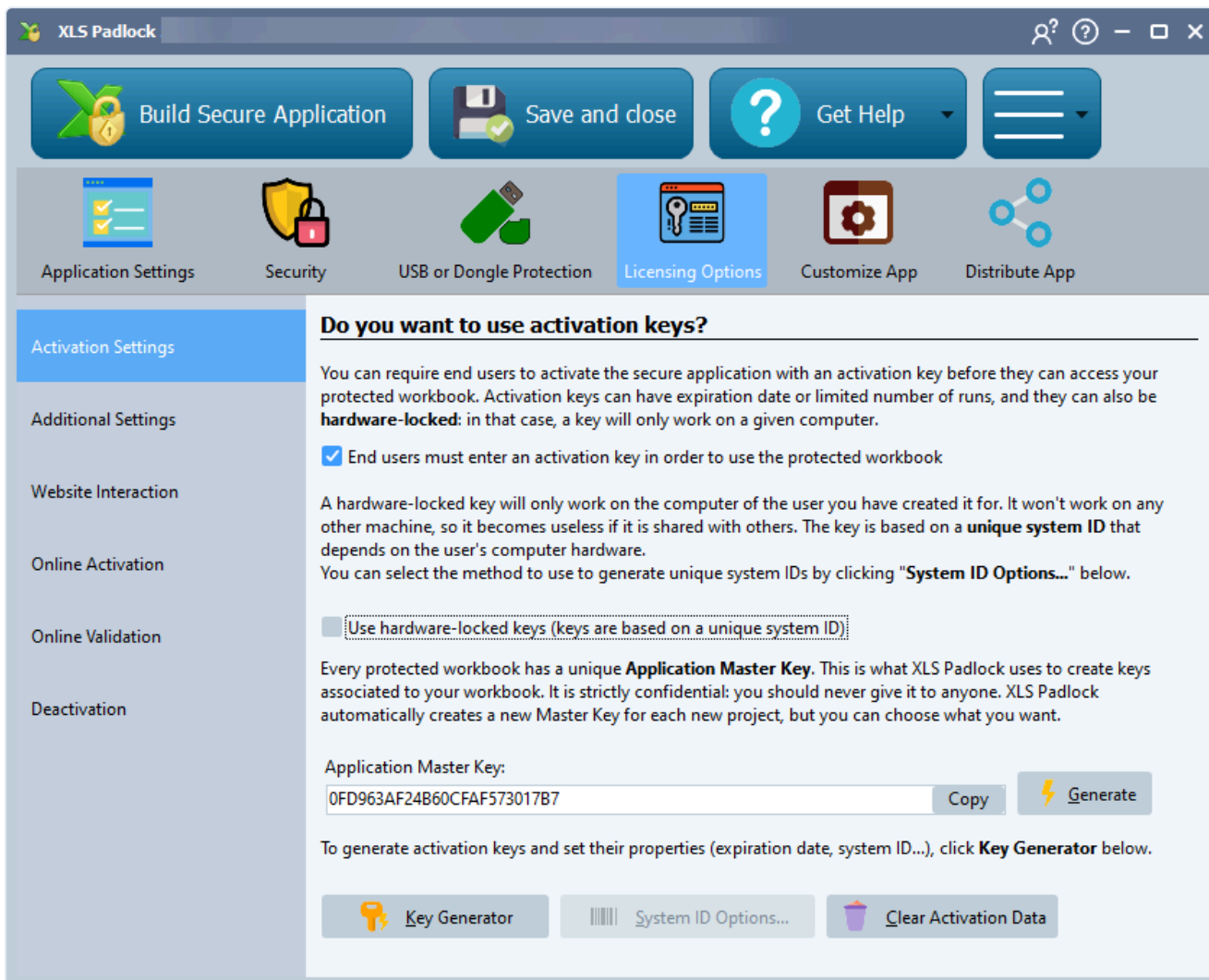
Activate Cancel

👉 Le chiavi di attivazione possono essere configurate con una [data di scadenza](#) o un [numero limitato di esecuzioni](#). Possono anche essere [bloccate sull'hardware](#), il che significa che una chiave funzionerà solo su un computer specifico.

SUGGERIMENTO

Oltre al [Key Generator](#) integrato, offriamo un generatore di chiavi autonomo in grado di creare migliaia di chiavi alla volta, oltre a un SDK per generatore di chiavi da integrare con il proprio server o sito web.

Infine, è possibile implementare le funzionalità di [attivazione online](#) e [convalida](#) per controllare a distanza il modo in cui gli utenti accedono alla cartella di lavoro e automatizzare la distribuzione delle chiavi di attivazione.



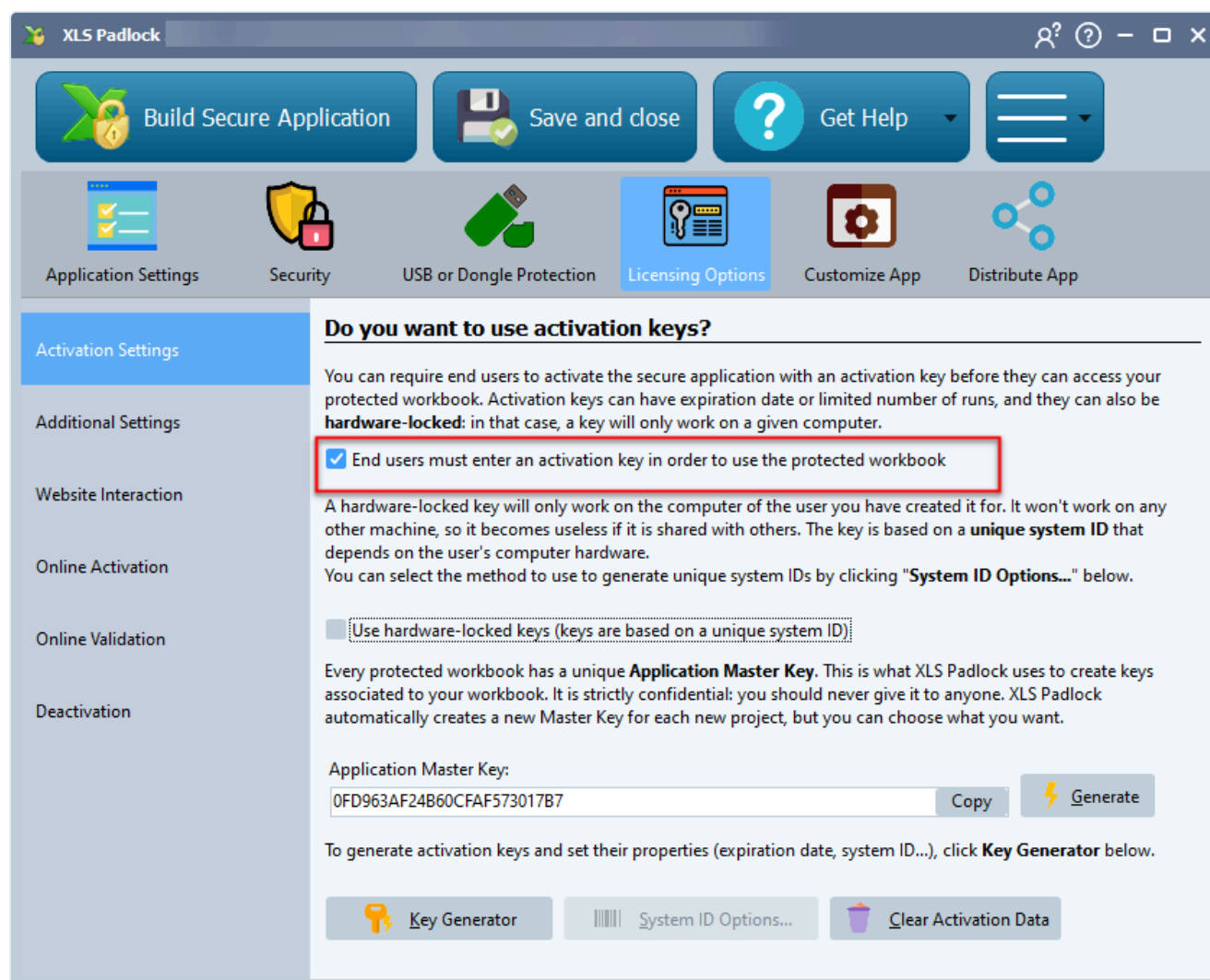
👉 Per iniziare, consultare il tutorial su [come configurare le chiavi di attivazione](#).

Configurare le chiavi di attivazione

Le **chiavi di attivazione** sono un potente sistema di licenza che consente di controllare chi può accedere alle cartelle di lavoro protette. Questa guida mostra come iniziare.

1. Abilitare le chiavi di attivazione

Nelle impostazioni di sicurezza di XLS Padlock, selezionare la casella **"End users must enter an activation key in order to use the protected workbook"** (gli utenti finali devono inserire una chiave di attivazione per utilizzare la cartella di lavoro protetta).

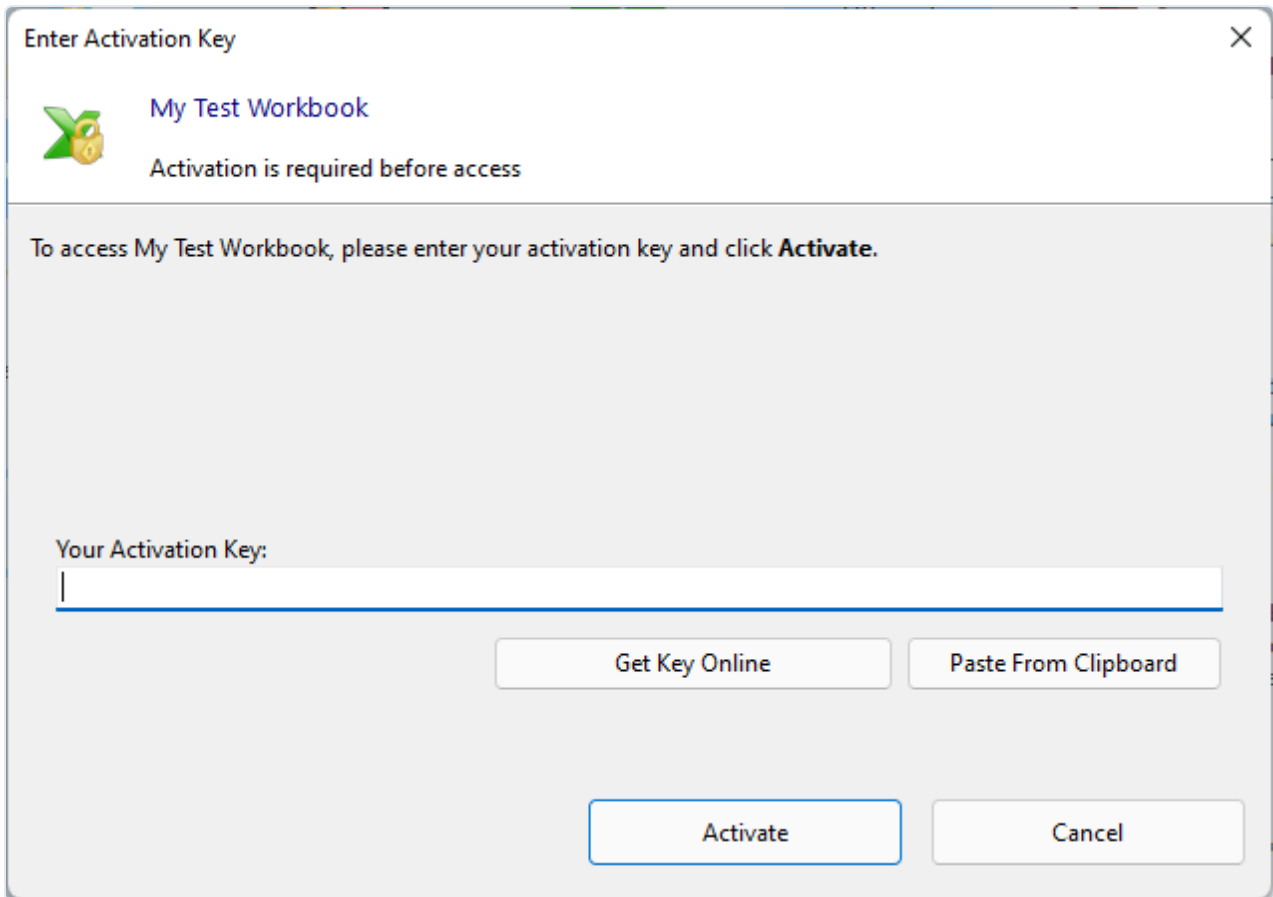


The screenshot shows the XLS Padlock application interface. At the top, there are buttons for 'Build Secure Application', 'Save and close', 'Get Help', and a menu icon. Below these are icons for 'Application Settings', 'Security', 'USB or Dongle Protection', 'Licensing Options' (which is selected), 'Customize App', and 'Distribute App'. The 'Licensing Options' panel is open, showing the 'Activation Settings' section. The checkbox 'End users must enter an activation key in order to use the protected workbook' is checked and highlighted with a red box. Below this, there is a section for 'Use hardware-locked keys (keys are based on a unique system ID)' with a radio button. Further down, there is an 'Application Master Key' field containing the text '0FD963AF24B60CFAF573017B7', a 'Copy' button, and a 'Generate' button. At the bottom, there are buttons for 'Key Generator', 'System ID Options...', and 'Clear Activation Data'.

Una volta fatto questo, ricompilare l'applicazione.

2. Provare l'applicazione

Quando si esegue l'applicazione appena compilata, verrà ora richiesto di inserire una chiave di attivazione prima di poter accedere alla cartella di lavoro:



3. Generare le chiavi di attivazione

Per creare le chiavi di attivazione, aprire il [Key Generator](#) in XLS Padlock.

Do you want to use activation keys?

You can require end users to activate the secure application with an activation key before they can access your protected workbook. Activation keys can have expiration date or limited number of runs, and they can also be **hardware-locked**: in that case, a key will only work on a given computer.

End users must enter an activation key in order to use the protected workbook

A hardware-locked key will only work on the computer of the user you have created it for. It won't work on any other machine, so it becomes useless if it is shared with others. The key is based on a **unique system ID** that depends on the user's computer hardware.

You can select the method to use to generate unique system IDs by clicking "**System ID Options...**" below.

Use hardware-locked keys (keys are based on a unique system ID)

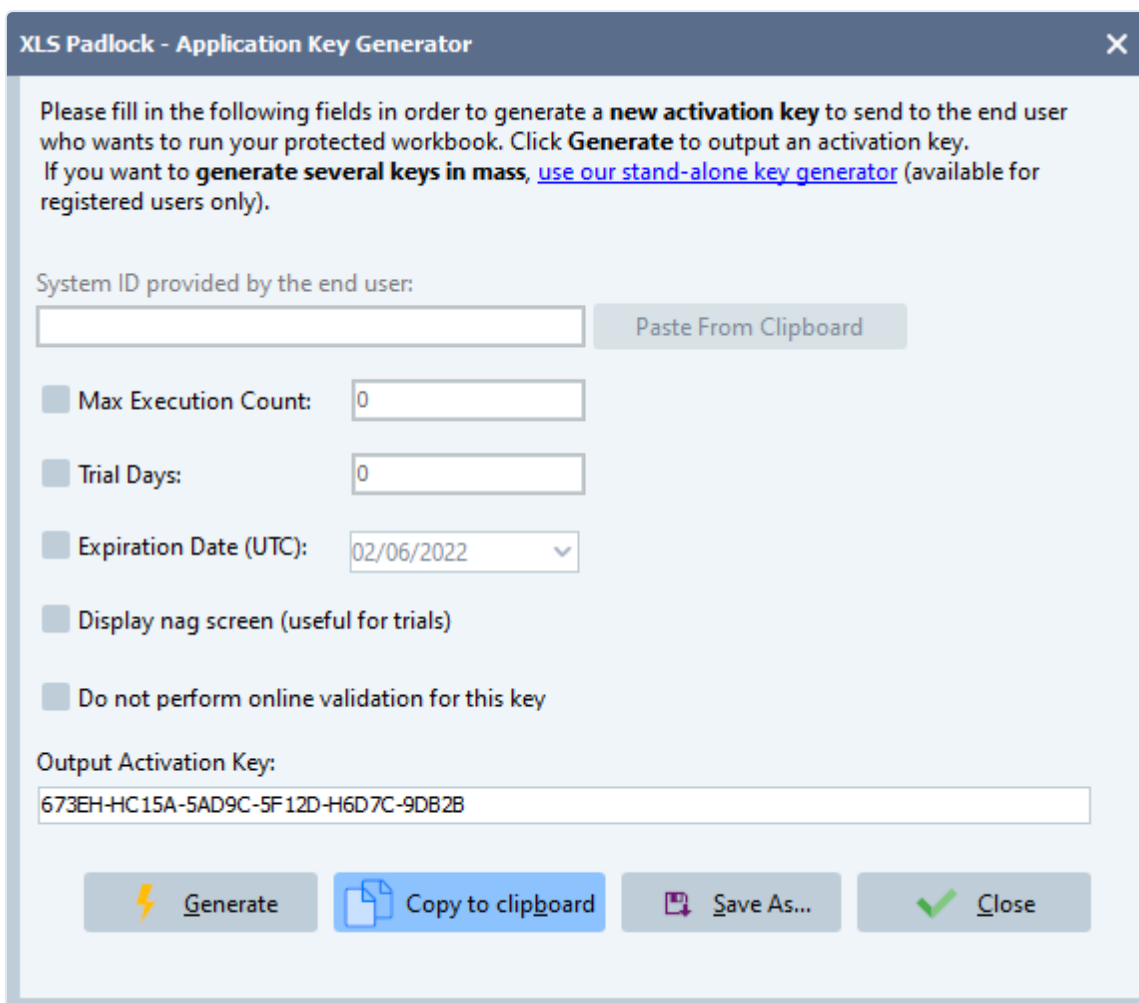
Every protected workbook has a unique **Application Master Key**. This is what XLS Padlock uses to create keys associated to your workbook. It is strictly confidential: you should never give it to anyone. XLS Padlock automatically creates a new Master Key for each new project, but you can choose what you want.

Application Master Key:

To generate activation keys and set their properties (expiration date, system ID...), click **Key Generator** below.

Se si desidera, è possibile [impostare restrizioni sulla chiave](#), come limitare il numero di esecuzioni o impostare una data di scadenza.

Fare clic sul pulsante **Generate** per creare istantaneamente una chiave. È quindi possibile copiare questa chiave negli appunti per inviarla al cliente.



XLS Padlock - Application Key Generator

Please fill in the following fields in order to generate a **new activation key** to send to the end user who wants to run your protected workbook. Click **Generate** to output an activation key. If you want to **generate several keys in mass**, [use our stand-alone key generator](#) (available for registered users only).

System ID provided by the end user:
 Paste From Clipboard

Max Execution Count:

Trial Days:

Expiration Date (UTC):

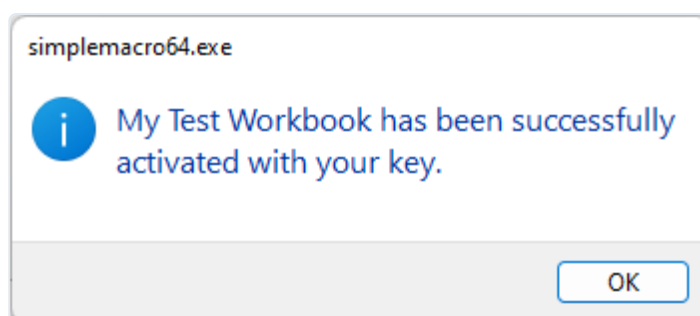
Display nag screen (useful for trials)

Do not perform online validation for this key

Output Activation Key:

⚡ Generate 📄 Copy to clipboard 💾 Save As... ✅ Close

Quando il cliente inserisce la chiave e fa clic su **Activate**, otterrà l'accesso alla cartella di lavoro. L'applicazione non richiederà più la chiave, a meno che non scada.



Automatizzare la generazione delle chiavi

Per facilitare la distribuzione delle chiavi, è disponibile per i clienti registrati un'applicazione [generatore di chiavi autonomo](#). Forniamo inoltre applicazioni web gratuite per generare le chiavi automaticamente tramite il proprio sito web.

Chiavi di attivazione con blocco hardware

Le **chiavi di attivazione con blocco hardware** sono un modo efficace per impedire la condivisione occasionale della cartella di lavoro protetta legando ogni licenza a un computer specifico. Questa guida mostra come configurarle.

Tutorial video disponibile

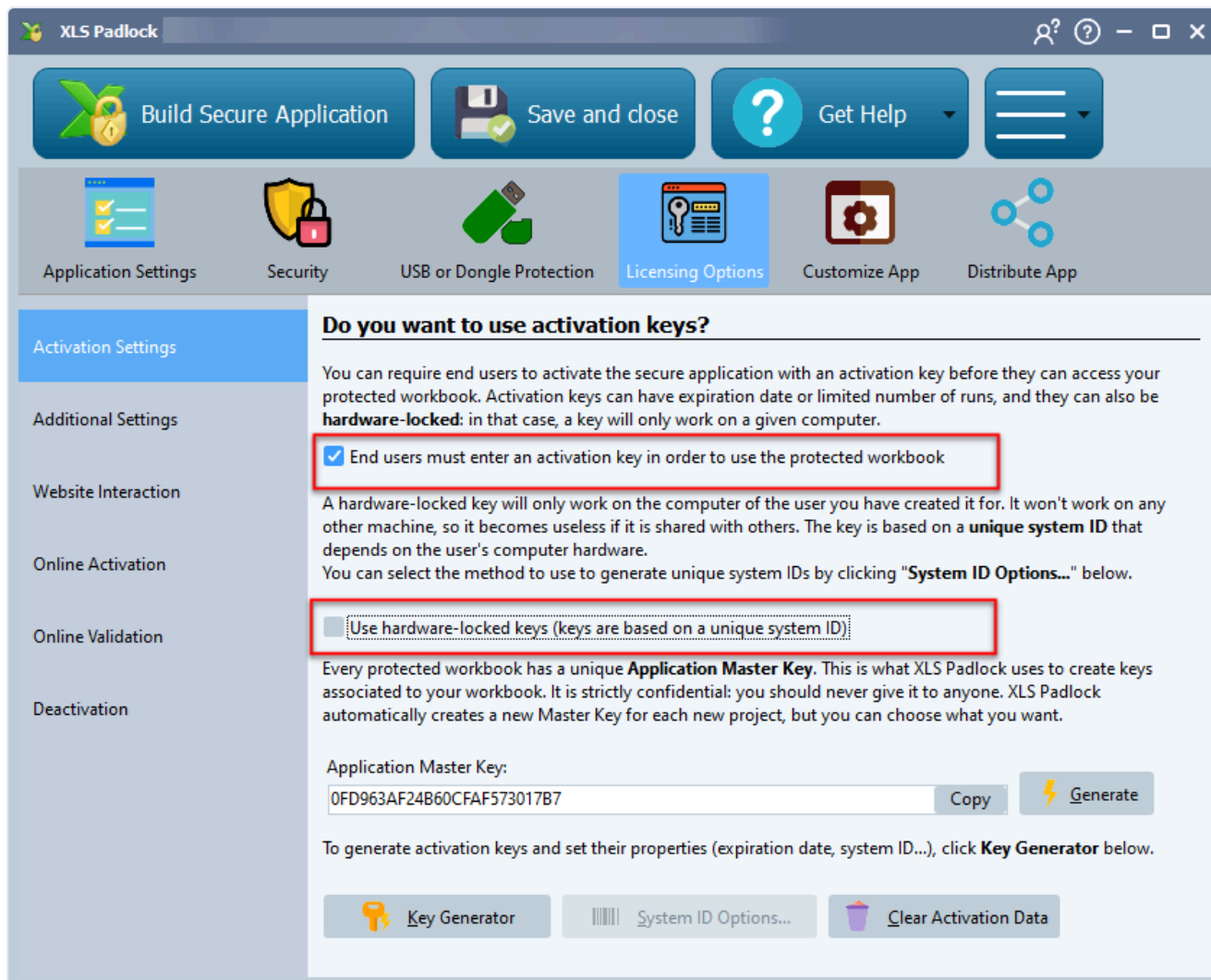
- [Guarda il nostro tutorial video sul blocco hardware dei tuoi fogli di calcolo Excel](#)
- [Guarda una dimostrazione dal vivo di questa funzionalità](#)

Informazioni sulle chiavi con blocco hardware

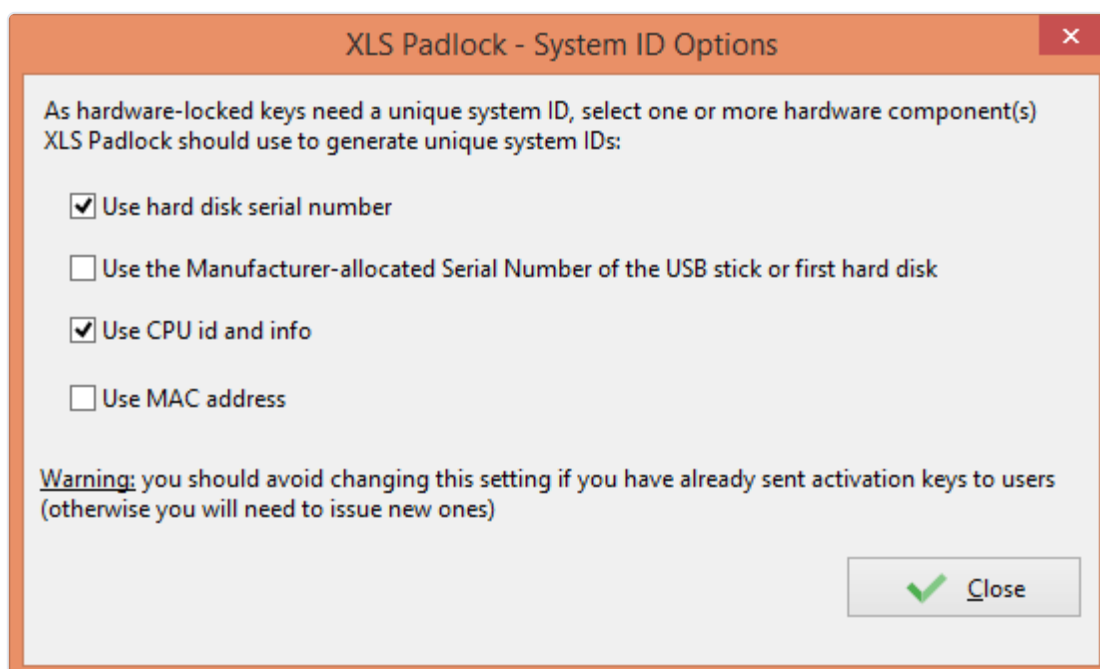
Una chiave con blocco hardware **funzionerà solo sul computer per cui è stata generata**. Si basa su un **System ID** univoco derivato dall'hardware del computer dell'utente, rendendo la chiave inutilizzabile se condivisa con altri.

1. Abilitare le chiavi con blocco hardware

Nelle impostazioni di XLS Padlock, selezionare sia "End users must enter an activation key..." sia "**Use hardware-locked keys**" (utilizzare chiavi con blocco hardware):



È possibile configurare facoltativamente quali componenti hardware vengono utilizzati per generare il System ID facendo clic su "**System ID Options...**". È importante non modificare queste opzioni dopo aver iniziato a distribuire l'applicazione.



Note sui componenti del System ID

- **CPU:** se la CPU non dispone di un identificatore di serie integrato, vengono utilizzate le sue informazioni generali. Per hardware identico, ciò potrebbe produrre lo stesso identificatore. È preferibile combinare più opzioni hardware.
- **MAC Address:** se gli utenti dispongono di più modi per connettersi a Internet (Wi-Fi, 4G, ecc.), il loro indirizzo MAC può cambiare, il che invaliderebbe la loro chiave.

Utilizzare l'impronta hardware avanzata

A partire da XLS Padlock 2026.0, la finestra di dialogo System ID Options espone anche una casella di controllo aggiuntiva: **"Use enhanced hardware fingerprint"** (utilizzare l'impronta hardware avanzata).

Quando questa opzione è abilitata, l'EXE protetto utilizza un calcolo rinforzato del System ID:

- La sorgente di blocco Windows legge l'identificatore univoco di installazione di Windows direttamente dal registro di sistema, anziché dalla scansione del volume del disco utilizzata dall'algoritmo legacy. Questo identificatore rimane stabile in caso di riformattazione del volume e di sostituzione del disco fisico. Solo una reinstallazione completa di Windows lo rigenera.
- Tutte le sorgenti hardware vengono normalizzate (spazi rimossi, maiuscole/minuscole canonicalizzate) e chiaramente delimitate all'interno del calcolo dell'impronta, eliminando una categoria di collisioni strutturali in cui due macchine distinte potevano occasionalmente produrre lo stesso System ID. La nuova logica protegge inoltre dai falsi negativi quando una sorgente diventa temporaneamente non disponibile (ad esempio quando un antivirus blocca le chiamate WMI), mantenendo stabile il System ID anziché farlo mutare.
- L'hash sottostante viene aggiornato a SHA-256 al posto del precedente MD5, rimuovendo un algoritmo obsoleto dall'impronta crittografica dell'EXE protetto.

Il formato del System ID visualizzato rimane invariato, sempre 14 caratteri esadecimale nella consueta forma `XXXX-XXXX-XXXX`, in modo che gli utenti finali e il flusso di lavoro di generazione delle chiavi non notino alcuna differenza.

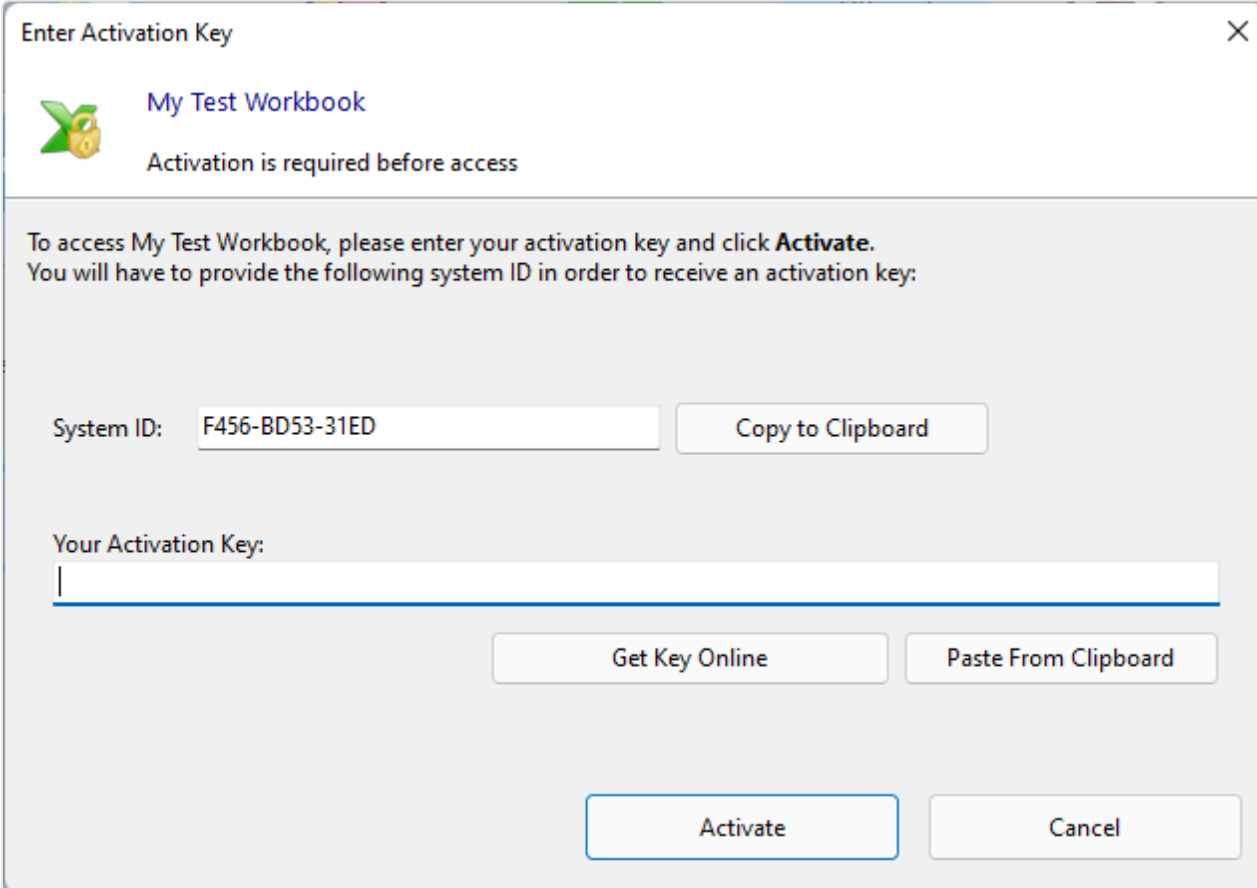
Comportamento predefinito:

- **Progetti esistenti** (salvati con XLS Padlock 2025.3 o precedente): l'opzione è **OFF** per impostazione predefinita. Questo preserva tutte le chiavi di attivazione già in circolazione: i System ID che i clienti hanno ottenuto dalle versioni precedenti continuano a corrispondere alle chiavi emesse.
- **Nuovi progetti** (creati con XLS Padlock 2026.0 e successivi): l'opzione è **ON** per impostazione predefinita.

⚠ Importante: modificare questa opzione su un progetto già distribuito ruoterà il System ID per ogni utente finale, invalidando tutte le chiavi di attivazione emesse finora. Abilitarla solo su un nuovo progetto, oppure coordinare la modifica con una riemissione della chiave di ogni cliente.

2. Come gli utenti ottengono il loro System ID

Quando un utente esegue l'applicazione per la prima volta, gli verrà richiesto di inserire una chiave di attivazione. Il suo System ID univoco verrà visualizzato in questa finestra. L'utente deve copiare questo identificatore e inviarlo a voi.



The screenshot shows a dialog box titled "Enter Activation Key" with a close button (X) in the top right corner. The dialog is for "My Test Workbook" and states "Activation is required before access". It contains the following elements:

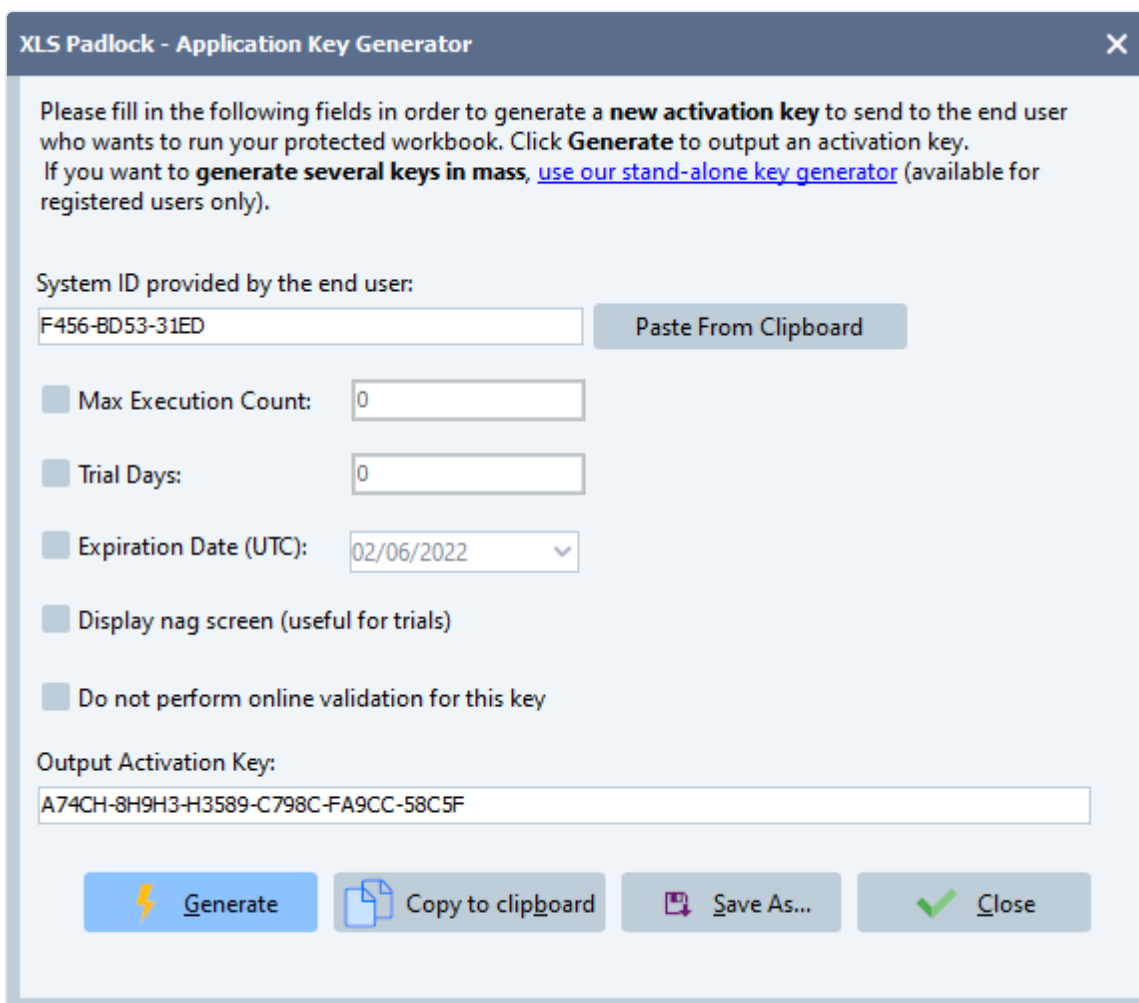
- A message: "To access My Test Workbook, please enter your activation key and click **Activate**. You will have to provide the following system ID in order to receive an activation key:"
- A "System ID" field containing the text "F456-BD53-31ED" and a "Copy to Clipboard" button to its right.
- A "Your Activation Key:" label above a large empty text input field.
- Two buttons below the input field: "Get Key Online" and "Paste From Clipboard".
- Two buttons at the bottom: "Activate" and "Cancel".

Automatizzare la consegna delle chiavi

Per evitare di scambiare manualmente i System ID, è possibile configurare un pulsante "[Get Key Online](#)" che invia l'utente al proprio sito web con il suo System ID, oppure utilizzare la funzionalità [Online Activation](#) completamente automatizzata.

3\ Generare la chiave di attivazione

Aprire il [Key Generator](#) in XLS Padlock. Incollare il System ID del cliente nell'apposito campo e fare clic su **Generate**. È quindi possibile copiare la chiave e inviarla al cliente.



XLS Padlock - Application Key Generator [Close]

Please fill in the following fields in order to generate a **new activation key** to send to the end user who wants to run your protected workbook. Click **Generate** to output an activation key. If you want to **generate several keys in mass**, [use our stand-alone key generator](#) (available for registered users only).

System ID provided by the end user:
F456-BD53-31ED Paste From Clipboard

Max Execution Count: 0

Trial Days: 0

Expiration Date (UTC): 02/06/2022

Display nag screen (useful for trials)

Do not perform online validation for this key

Output Activation Key:
A74CH-8H9H3-H3589-C798C-FA9CC-58C5F

⚡ Generate 📄 Copy to clipboard 💾 Save As... ✅ Close

Quando il cliente inserisce la chiave, l'applicazione verrà attivata e non richiederà più la chiave (a meno che non abbia una data di scadenza). Poiché la chiave è legata al suo System ID, non funzionerà su nessun altro computer.

👉 Vuoi automatizzare la creazione di chiavi di attivazione con blocco hardware? [Scopri di più sull'attivazione online.](#)

Attivazione online

Come funziona

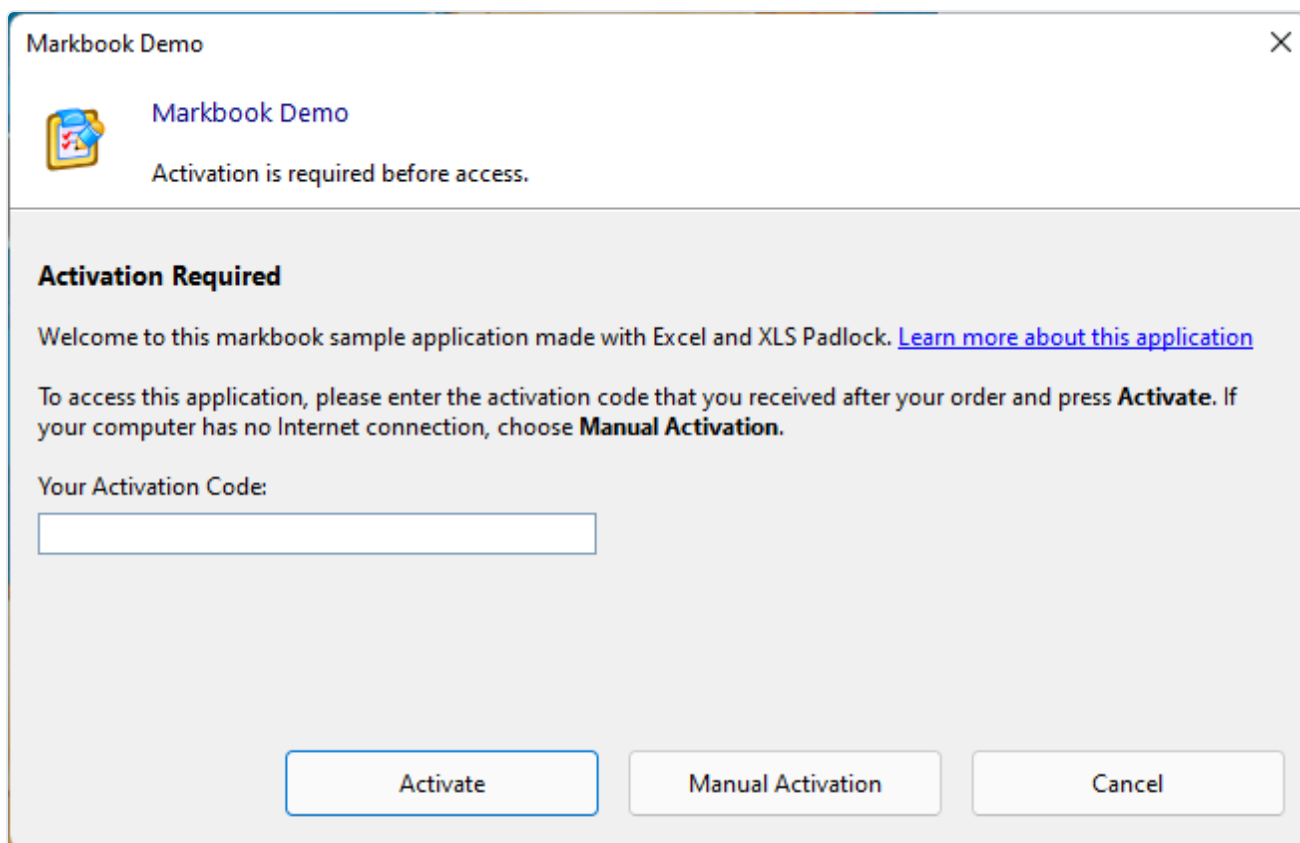
L'attivazione online automatizza il recupero delle [chiavi di attivazione](#) tramite internet. L'applicazione protetta comunica con il proprio server web per scaricare direttamente una chiave di attivazione, eliminando la necessità per gli utenti di inserirla manualmente.

Kit lato server richiesto

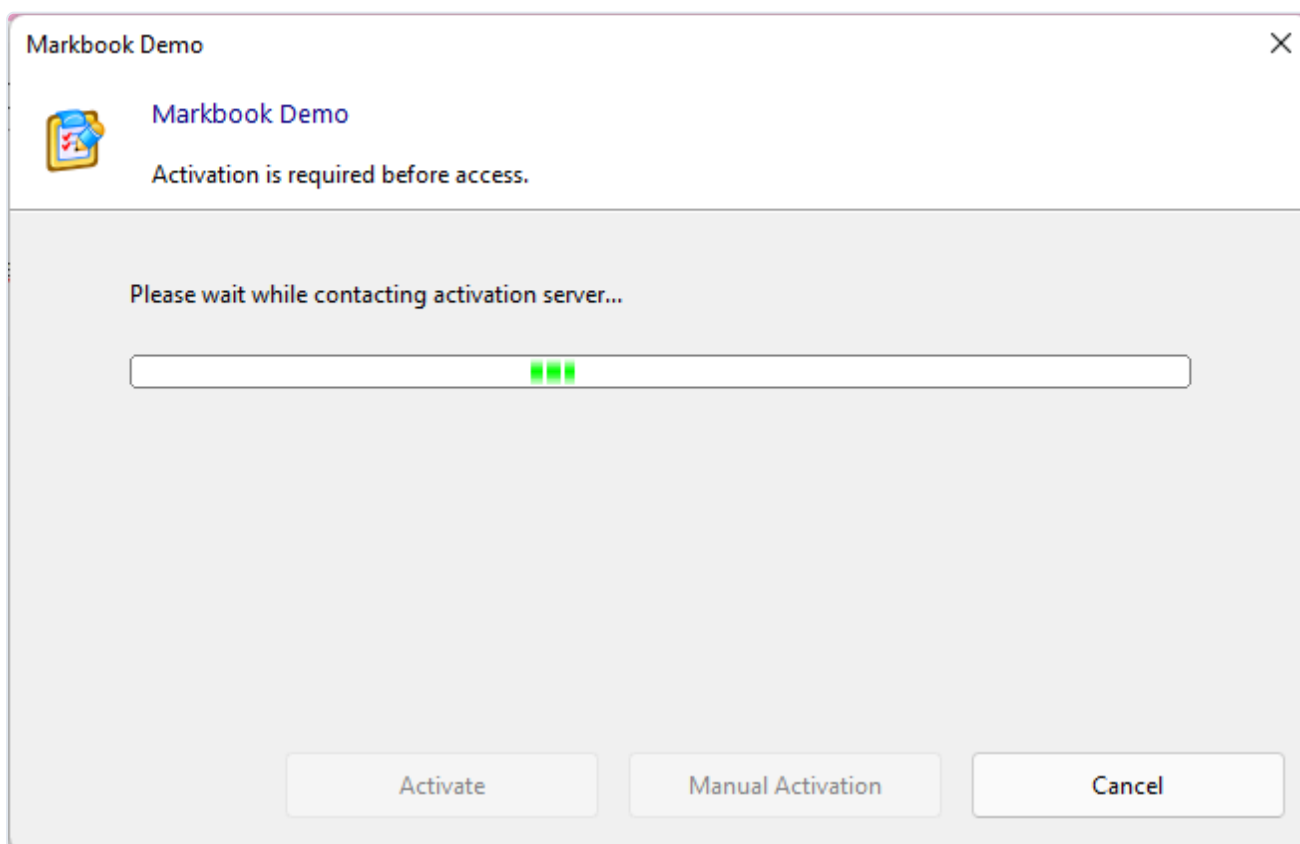
Per utilizzare l'attivazione online, è necessario installare uno dei nostri kit lato server sul proprio server web, come il XLS Padlock Activation Kit, il WooCommerce Integration Kit o il FastSpring Subscription kit. È possibile scaricare questi kit dalla propria [pagina account](#).

Se si opta per il protocollo di attivazione 2026 (firma delle risposte Ed25519, vedere più avanti), assicurarsi che il kit sia nella versione 2026.0 o successiva. Il kit rileva automaticamente il protocollo per ogni richiesta, in modo che le cartelle di lavoro precedenti continuino a funzionare con la stessa installazione del kit.

👉 Quando un utente finale avvia l'applicazione, compare una finestra di dialogo che lo invita a procedere con l'attivazione. Questa finestra di dialogo sostituisce la finestra di dialogo standard "Enter Activation Key" ed è completamente personalizzabile. È possibile aggiungere campi propri per raccogliere dati dall'utente (come un numero d'ordine o un indirizzo email), che verranno inviati al proprio server web. Il server convalida quindi queste informazioni e, in caso di esito positivo, rinvia una chiave di attivazione all'applicazione.



Quando l'utente fa clic su **Activate**, i dati vengono inviati al server di attivazione:



Dopo un'attivazione riuscita, viene visualizzato un messaggio di conferma e l'applicazione si riavvia. Se si verifica un errore, comparirà una finestra di messaggio che consente all'utente di riprovare.

Configurazione

👉 Per abilitare l'attivazione online, è necessario configurare le seguenti opzioni:

Base Activation URL

Indicare l'URL completo del kit di attivazione installato sul proprio server. Ad esempio, se il kit è stato installato in una sottocartella denominata "activation", l'URL sarà

```
https://www.yourdomain.com/activation/getactivation/.
```

Utilizzare HTTPS

Le connessioni sicure tramite TLS/SSL sono supportate. È necessario utilizzare sempre URL che iniziano con `https://`.

⚠️ Lasciare il campo vuoto se ****non**** si desidera utilizzare l'attivazione online.

Identificazione del cliente: chiave di sicurezza o coppia di chiavi Ed25519

A partire da XLS Padlock 2026.0, è possibile scegliere tra due regimi di protocollo di attivazione tramite l'opzione **Compatibility mode for pre-2026 activation kits** (modalità di compatibilità per i kit di attivazione antecedenti al 2026):

- **Compatibility mode selezionata** (impostazione predefinita, consigliata se il proprio server di attivazione è antecedente alla versione 2026): il campo GUID **Security Private Key legacy** identifica l'applicazione presso il kit di attivazione. Il protocollo è basato su righe con risposte non firmate. Questo è il comportamento di tutte le versioni di XLS Padlock antecedenti alla 2026.0.
- **Compatibility mode deselezionata** (richiede la versione 2026 del XLS Padlock Activation Kit, del WooCommerce Integration Kit o del FastSpring Subscription Kit): il campo legacy è nascosto e subentra una **Ed25519 keypair** (coppia di chiavi Ed25519) specifica per progetto. Fare clic sul pulsante **Generate keypair** (genera coppia di chiavi) nella pagina Online Activation per produrre una nuova coppia di chiavi. La chiave pubblica viene incorporata nella cartella di lavoro protetta; la chiave privata viene mostrata in una finestra di dialogo a visualizzazione unica, in modo da poterla incollare nella configurazione del proprio kit di attivazione (`xlspadlocksignkey` in `config.ini`).

Se si perde la chiave privata, fare clic su **Show config.ini snippet** nella stessa pagina per visualizzarla di nuovo. La chiave è conservata nel proprio file di progetto `.xplp`. Trattare il file `.xplp` come riservato: non inserirlo nel controllo di versione, non inviarlo via email per richiedere assistenza.

Il protocollo 2026 utilizza una busta di richiesta JSON e verifica una firma Ed25519 separata su ogni risposta del server di attivazione. Questo protegge i clienti da false risposte di "avvenuta attivazione" o da messaggi di errore di phishing che un attaccante a livello di rete potrebbe altrimenti iniettare aggirando TLS (proxy aziendale con iniezione di CA, antivirus malevolo, certificato radice compromesso).

Allow Manual Activation if No Internet Connection

Alcuni utenti potrebbero non disporre di una connessione internet attiva. Per consentire loro di attivare manualmente, abilitare l'opzione **"Allow Manual Activation if No Internet Connection"**.

Questo metodo manuale funziona esattamente come le chiavi di registrazione standard. In questo caso, è necessario essere pronti a gestire le richieste di attivazione degli utenti offline.

Avvertenza

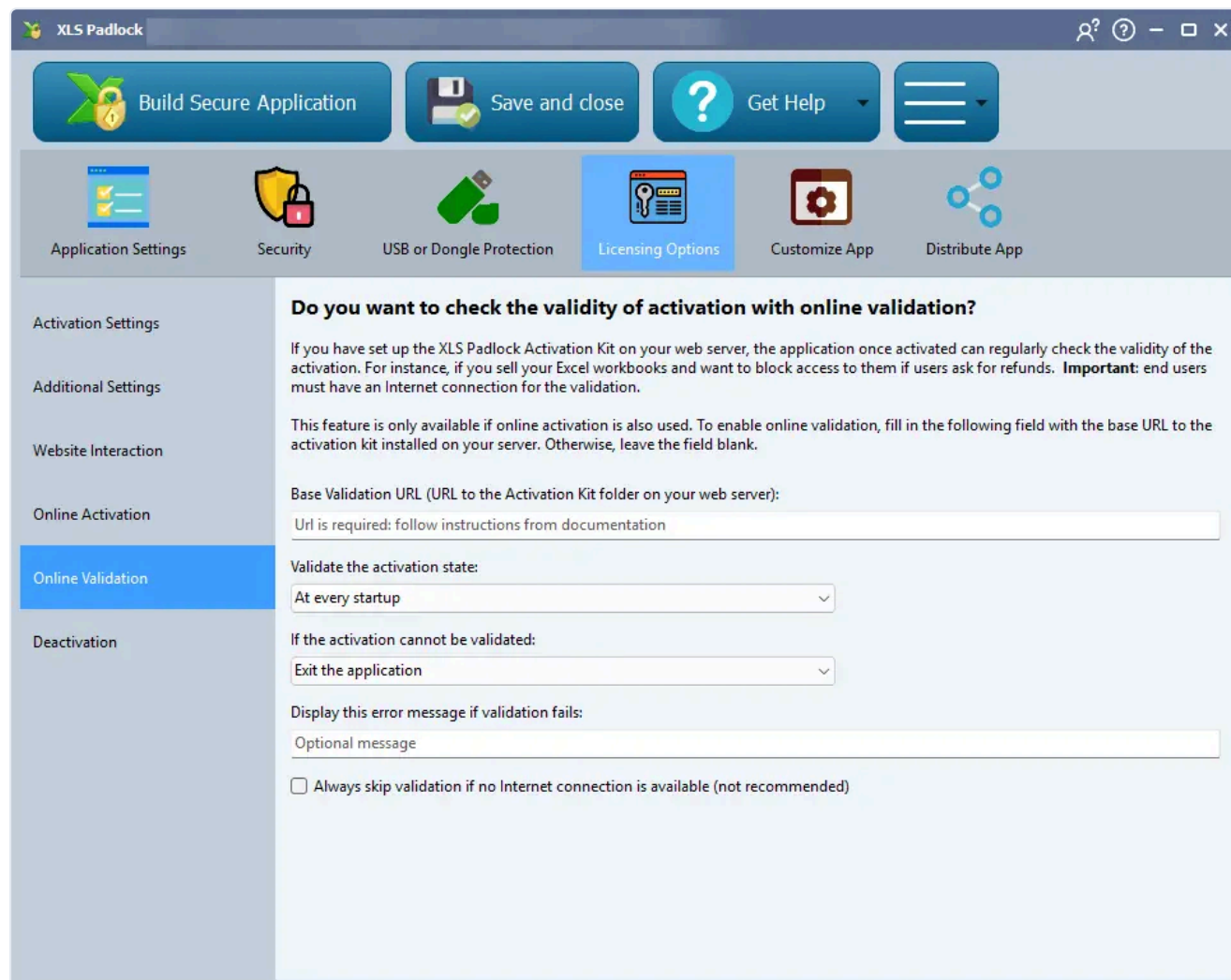
Se si accettano le attivazioni manuali, assicurarsi di generare chiavi di attivazione che **non eseguano la validazione online** (opzione nel [generatore di chiavi](#)). In caso contrario, la validazione non andrà a buon fine!

Vedere anche

- [Come vendere cartelle di lavoro Excel con abbonamenti](#)
- [Mostrare un pulsante "Purchase Online" nella schermata di promemoria](#)

Convalida online

La validazione online offre un **controllo a distanza** sulle applicazioni attivate. Quando viene utilizzata con uno dei nostri kit di attivazione lato server (come il XLS Padlock Activation Kit, il WooCommerce Kit o il [kit di abbonamento FastSpring](#)), l'applicazione può verificare periodicamente presso il proprio server che la sua attivazione rimanga valida.



Avvertenza

Questa funzionalità richiede che l'[attivazione online](#) sia prima abilitata e configurata.

Questo è utile in diversi scenari:

- È possibile disattivare a distanza un'applicazione se un cliente richiede un rimborso.
- È possibile imporre un accesso basato su abbonamento, in cui l'applicazione smette di funzionare se l'abbonamento scade.

Come funziona

- La validazione online utilizza lo stesso token univoco generato durante l'attivazione online iniziale per identificare il dispositivo del cliente.
- La chiave di attivazione stessa non viene mai trasmessa tramite internet durante la validazione.
- Per eseguire il controllo di validazione è necessaria una connessione internet attiva.

Configurazione

👉 È necessario configurare la **Base Validation URL** affinché punti al proprio script di validazione lato server. Si tratta generalmente dell'URL del XLS Padlock Activation Kit, del XLS Padlock WooCommerce Integration Kit o del kit di abbonamento FastSpring sul proprio server web. Ad esempio, se il kit di attivazione è stato installato in una sottocartella denominata "activation", l'URL sarà `https://www.yourdomain.com/activation/dovalidation/`.

Utilizzare HTTPS

Le connessioni sicure tramite TLS/SSL sono supportate. È necessario utilizzare sempre URL che iniziano con `https://`.

⚠️ Lasciare il campo vuoto se ****non**** si desidera utilizzare la validazione online.

Frequenza di validazione

Scegliere quando l'applicazione deve eseguire una validazione: a ogni avvio, in modo casuale, ogni X giorni oppure ogni X esecuzioni. È necessario specificare il valore di X dove richiesto.

Avvertenza

Dopo l'attivazione, al lancio successivo è richiesta una validazione iniziale. Dopo questo primo controllo, l'applicazione rispetterà la frequenza di validazione selezionata.

Azione in caso di esito negativo della validazione

Definire cosa deve fare l'applicazione se la validazione online ha esito negativo: * **Exit the application:** l'applicazione si chiuderà immediatamente. * **Blacklist activation key:** la chiave attuale verrà invalidata e l'utente sarà invitato a inserirne una nuova. Se l'utente reinserisce la stessa chiave, la validazione verrà tentata di nuovo. * **Do nothing:** l'applicazione continuerà a funzionare. È possibile utilizzare le estensioni VBA di XLS Padlock per verificare lo stato di validazione e implementare una logica personalizzata.

Se l'attivazione non può essere validata

È possibile visualizzare un messaggio di errore personalizzato per l'utente in caso di esito negativo della validazione, indicandogli come procedere.

Ignorare la validazione se offline

È possibile consentire all'applicazione di ignorare il controllo di validazione se non viene rilevata alcuna connessione internet. Questo non è generalmente consigliato per motivi di sicurezza.

Avvertenza

Se questa opzione è disabilitata, gli utenti devono disporre di una connessione internet attiva per il processo di validazione. In caso contrario, la validazione non andrà a buon fine.

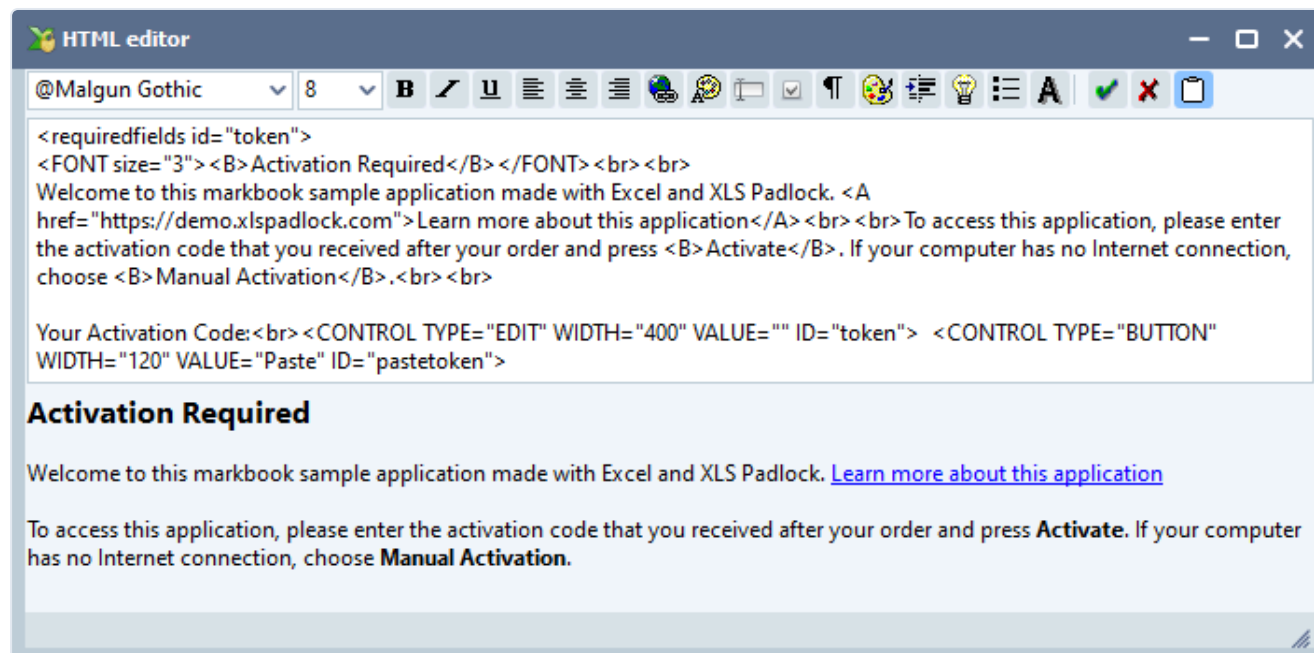
VBA: recuperare le informazioni di abbonamento/licenza

Dopo una validazione online riuscita, è possibile recuperare i dati personalizzati inviati dal proprio server utilizzando la chiamata VBA `XLSPadLock.PLEvalVar("ValidationAddServerData")`. Questo è utile per trasmettere alla cartella di lavoro informazioni specifiche dell'abbonamento.

Editor del modulo di registrazione

L'editor del modulo di registrazione consente di personalizzare il testo che appare nella [finestra di dialogo di attivazione online](#). L'editor supporta l'HTML di base, quindi è possibile utilizzare tag come `` , `
` , `` , ecc.

XLS Padlock fornisce un editor HTML di base con un'anteprima in tempo reale del testo della finestra di dialogo:



Controlli personalizzati

È possibile aggiungere campi personalizzati per richiedere informazioni aggiuntive agli utenti, che verranno inviate al proprio server web. È inoltre possibile aggiungere un pulsante per incollare il testo nel campo del token.

- **Paste Button (pulsante Incolla):** `<CONTROL TYPE="BUTTON" WIDTH="120" VALUE="Paste" ID="pastetoken">`
- **Text Input (campo di testo):** `<CONTROL TYPE="EDIT" WIDTH="400" VALUE="" ID="token">`

Campi obbligatori

È possibile rendere obbligatori dei campi elencandone gli ID in un tag `<requiredfields>`. Ad esempio, per rendere obbligatori i campi "token" e "name":

```
<requiredfields id="token,name">
```

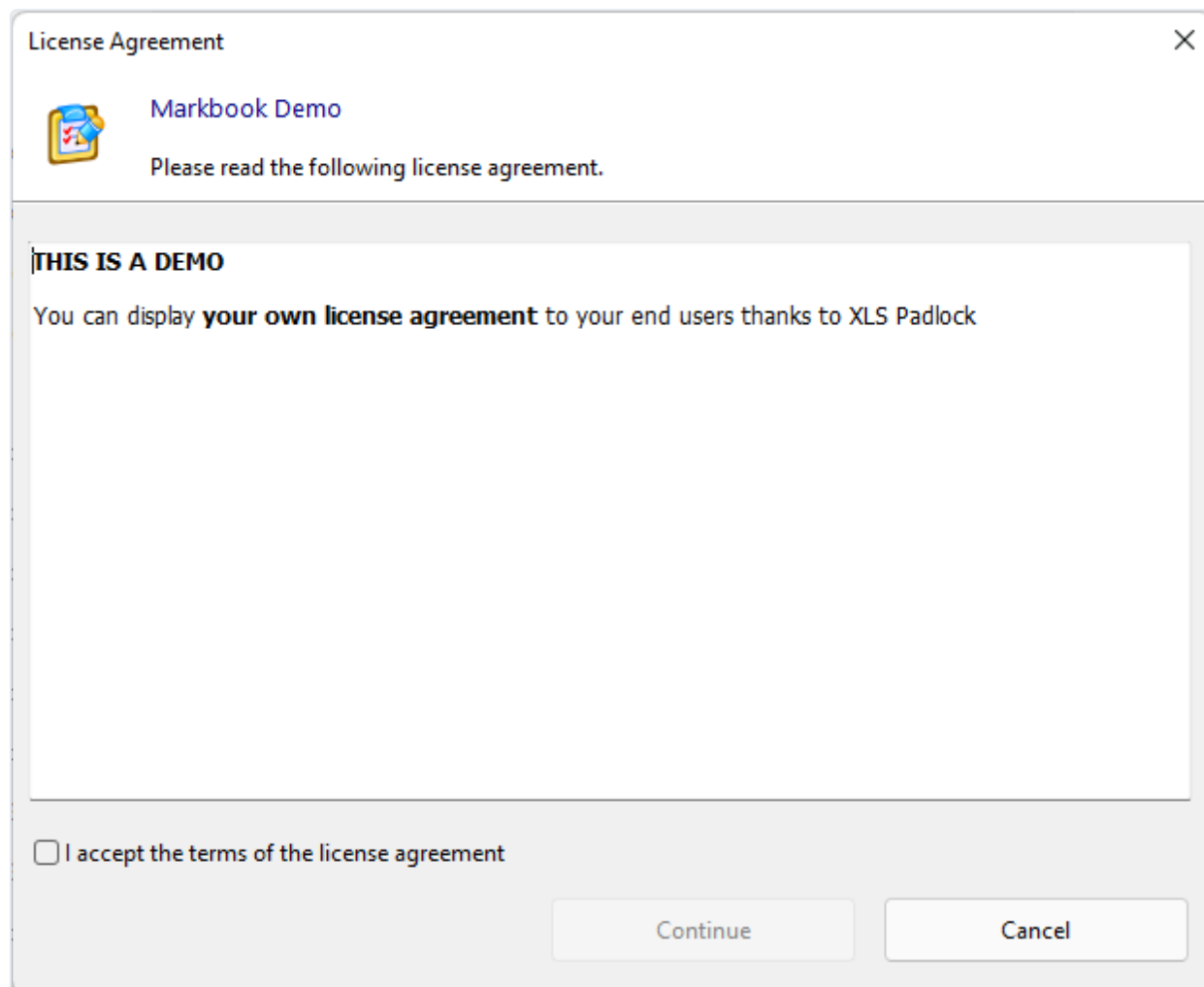
Esempio completo

```
<requiredfields id="token">
<FONT size="3"><B>Activation Required</B></FONT><br><br>
Welcome to this application.<br><br>
To access this application, please enter the activation code that you received after your order
Your Activation Code:<br>
<CONTROL TYPE="EDIT" WIDTH="400" VALUE="" ID="token">&nbsp; <CONTROL TYPE="BUTTON" WIDTH="120"
```

Mostrare il contratto di licenza

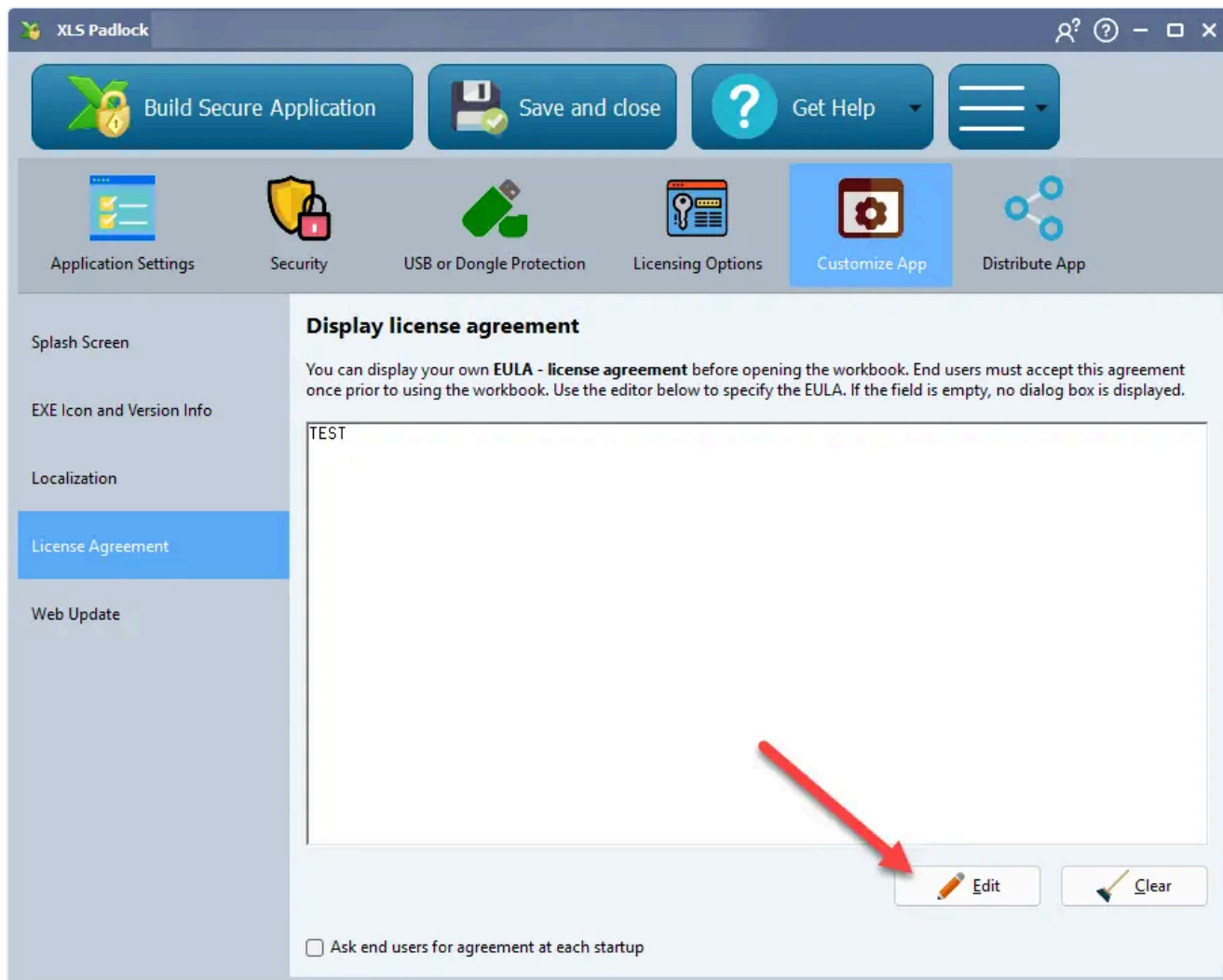
È possibile richiedere agli utenti di accettare un **End User License Agreement (EULA)** (contratto di licenza con l'utente finale) prima di poter aprire la cartella di lavoro protetta.

La cartella di lavoro compilata visualizzerà una finestra di dialogo come illustrato di seguito:



Quando l'utente seleziona "I accept the terms of the license agreement", il pulsante "Continue" diventa attivo, permettendogli di proseguire.

👉 Per aggiungere il proprio EULA, fare clic su **Edit** per aprire un editor di testo formattato in questa pagina:



È possibile caricare il contenuto da un file RTF o incollare il testo direttamente da un'applicazione come Microsoft Word.

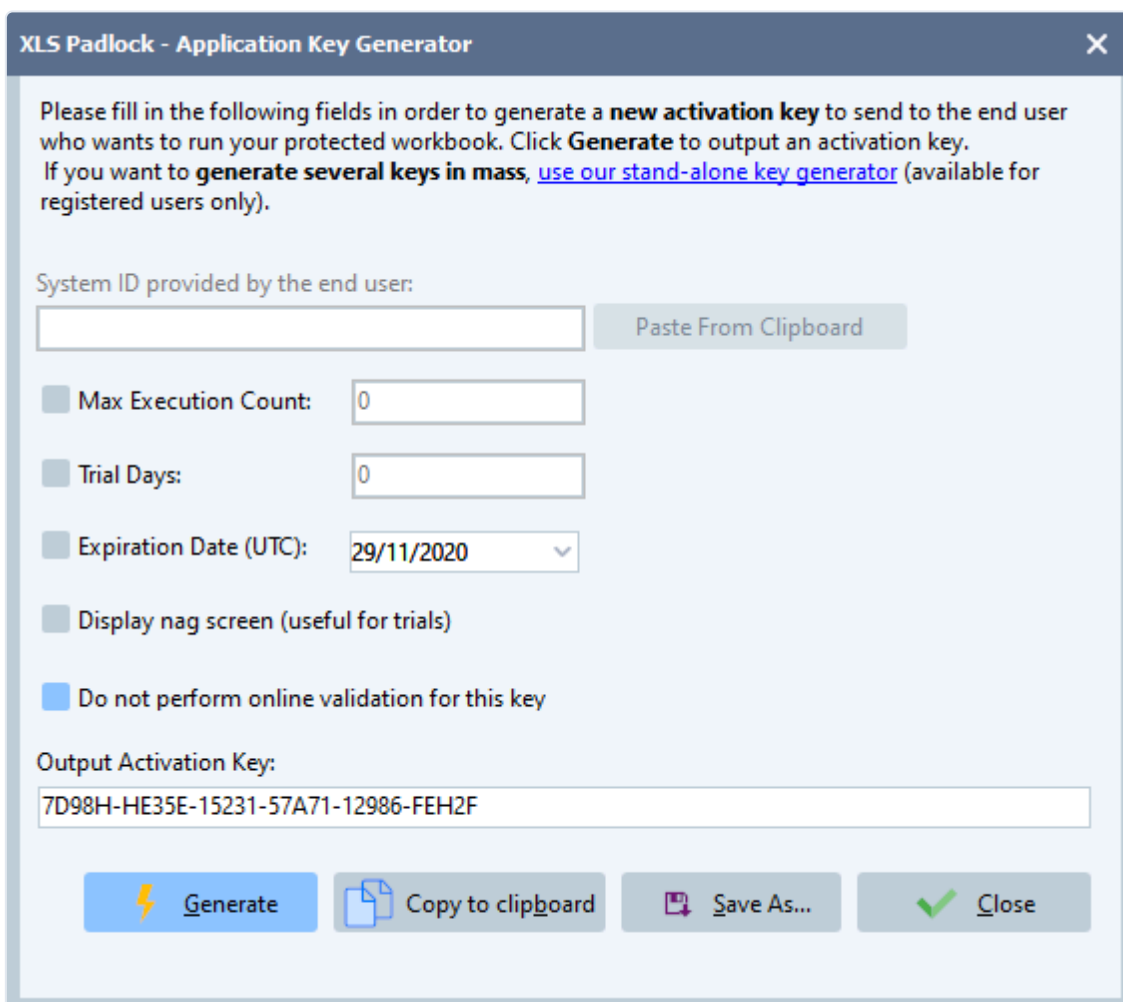
NOTA

Se il contenuto dell'EULA viene lasciato vuoto, non verrà visualizzata alcuna finestra di dialogo.

i Per mostrare l'EULA a ogni avvio dell'applicazione, abilitare l'opzione **Ask end users for agreement at each startup**. Per impostazione predefinita, gli utenti devono accettarlo una sola volta.

Generatore di chiavi (portatile e server)

XLS Padlock include un **Key Generator** (generatore di chiavi) integrato per creare chiavi di attivazione per le applicazioni delle cartelle di lavoro.



- **Per le chiavi di attivazione semplici**, è sufficiente fare clic sul pulsante **Generate** per creare istantaneamente una chiave.
- **Per le chiavi con blocco hardware**, inserire il System ID fornito dall'utente e quindi fare clic su **Generate**. La chiave risultante sarà vincolata a quello specifico computer.

È possibile copiare la chiave generata negli appunti per inviarla al cliente, oppure salvarla come file `.txt` per i propri archivi.

Le chiavi sono specifiche della cartella di lavoro

Il generatore di chiavi crea chiavi di attivazione solo per il progetto di cartella di lavoro attualmente aperto. Le chiavi generate per una cartella di lavoro non funzioneranno con un'altra, a meno che non condividano la stessa [Application Master Key](#).

SDK del generatore di chiavi

Offriamo un Key Generator SDK gratuito (un'applicazione web basata su PHP) che è possibile installare sul proprio sito web. Questo SDK consente di automatizzare la generazione e la distribuzione delle chiavi di attivazione per le proprie cartelle di lavoro.

A differenza del [generatore di chiavi integrato](#), che crea le chiavi manualmente all'interno di XLS Padlock, questo SDK viene eseguito sul proprio server per emettere le chiavi automaticamente.

Per una soluzione di licenza completamente automatizzata, valutare l'abbinamento dell'SDK con la [funzionalità di attivazione online](#).

👉 È possibile scaricare il Key Generator SDK dalla propria [pagina dell'account](#).

Generatore di chiavi autonomo

XLS Padlock offre un generatore di chiavi autonomo per i clienti registrati. Questo strumento consente di [creare chiavi di attivazione](#) per le proprie cartelle di lavoro protette senza dover aprire Excel o il componente aggiuntivo XLS Padlock.

Please fill in the following fields in order to generate **new activation keys** for end users who want to run your protected workbook.

First, enter the same **Application Master Key** as for your workbook in the XLS Padlock window. Do not share this key with others since it is used to generate activation keys for your protected workbook.

Click **Generate** to output one or more activation keys at once.

Application Master Key:
D... Paste From Clipboard

Use hardware-locked keys (keys are based on a unique system ID)

System ID provided by the end user:
... Paste From Clipboard

OR text file with all system IDs (one per line):
... Browse...

Max Execution Count: 0

Trial Days: 0

Expiration Date (UTC): 19/12/2016

Display nag screen

Generate Key(s) Number of Keys: 100

Load/Save Profile: ... Save Load

Copy to clipboard Save As... Close

1	EH4HH-H7A18-F9CA7-C9FBE-C67C6-91F2B
2	2B54H-HFEAB-647H6-762B3-CH5H4-CE75E
3	8DE6H-H899H-9HF8F-25EF7-184A1-E662H
4	DE6CH-H9316-16AE1-6H1B6-78DFF-B7328
5	4EHFH-H454F-4F681-81E68-C8BB5-DEB5B
6	E39BH-H23A2-E611E-6HFFC-EF2FB-BB622
7	7913H-H4C6H-3B4D7-3D54D-55738-D2H2E
8	3AF7H-H2A7H-1A51A-53685-1CA55-D2621
9	CEA7H-HDA27-A4E66-77H77-5B98F-36H58
10	1H64H-HC2HD-2D222-DDHA3-HF9DB-1825F
11	B528H-H935F-7DFF7-D4988-D56DC-44A25
12	7743H-H7B58-58A9A-5HB64-42A5B-46627
13	E287H-H6DA5-A56D6-4F9H3-75646-EE528
14	6CAAH-H41F9-F9HC1-A1222-BA92F-6E555
15	6E41H-H5EB7-H9587-AH158-697A2-4E12B
16	B27FH-H15E1-E118F-59E18-2BHA4-FA124
17	4538H-H5H5E-82FCB-67A74-91FAA-B7B25
18	AH5HH-H7E17-1D865-C45H1-FC5CC-3FE2H
19	E542H-H6416-99AAC-FC2FB-B7116-1D62E
20	4BA9H-H2C2C-D9163-3HFB4-3333B-37H2F
21	38ECH-HC3AF-363C3-654A1-776A9-4C956
22	3232H-H18A1-F9451-392C8-99134-6F826

Per una gestione più semplice, il generatore di chiavi autonomo consente inoltre di salvare le impostazioni sotto forma di **profili**.

👉 È possibile scaricare il generatore di chiavi autonomo dalla propria [pagina dell'account](#).

Restrizioni sulle chiavi

Quando si generano le [Activation Keys](#) con XLS Padlock, è possibile **applicare diverse restrizioni** tramite le seguenti opzioni:

XLS Padlock - Application Key Generator

Please fill in the following fields in order to generate a **new activation key** to send to the end user who wants to run your protected workbook. Click **Generate** to output an activation key.
If you want to **generate several keys in mass**, [use our stand-alone key generator](#) (available for registered users only).

System ID provided by the end user:

Max Execution Count:

Trial Days:

Expiration Date (UTC):

Display nag screen (useful for trials)

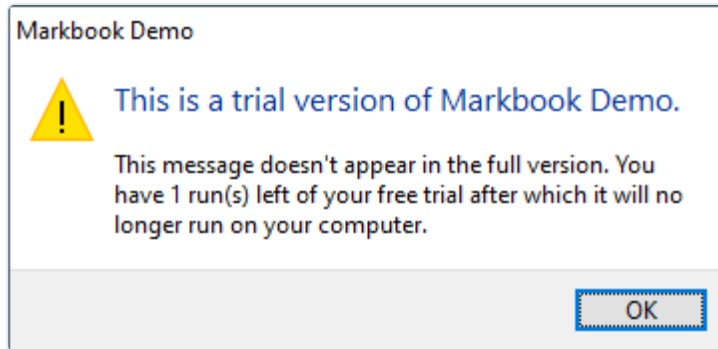
Do not perform online validation for this key

Output Activation Key:

☞ Limitare il numero di volte in cui l'applicazione può essere eseguita abilitando **"Max Execution Count"** e inserendo il numero desiderato.

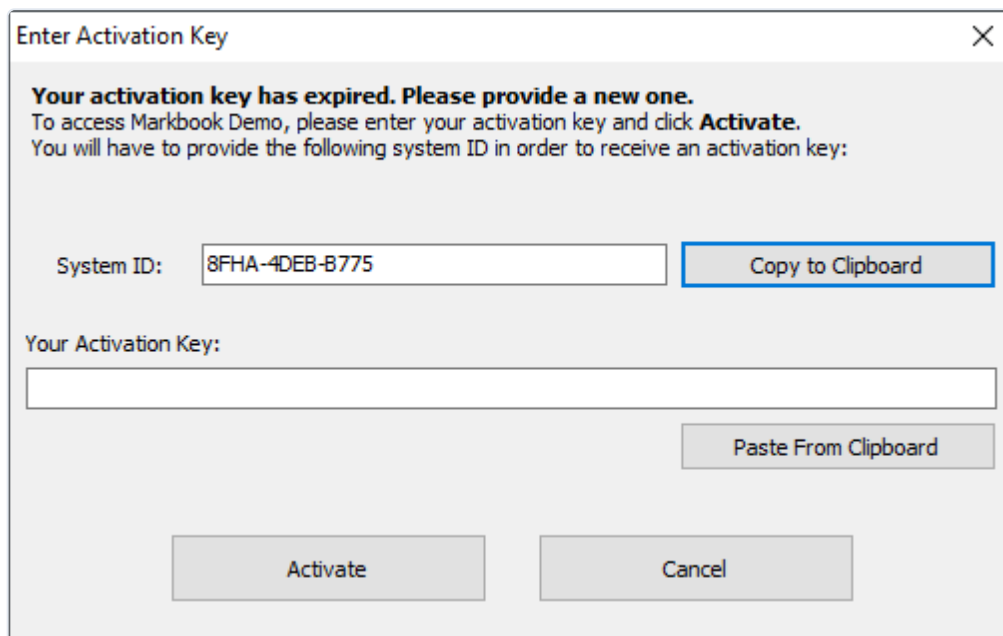
☞ Far scadere la chiave dopo un numero preciso di giorni (abilitare **"Trial Days"**) oppure a una data precisa (abilitare **"Expiration Date"**).

☞ Visualizzare una finestra di dialogo di promemoria (o "nag screen") all'avvio abilitando **"Display Nag Screen"**. Questa funzione è [utile per le versioni di prova](#), poiché informa l'utente del numero di giorni o di esecuzioni rimanenti.



Infine, se si utilizza la [validazione online](#) ed è necessario esentare una chiave specifica da questa verifica (ad esempio, per un utente in un ambiente offline), è possibile abilitare "**Do not perform online validation for this key**". Questa opzione è consigliata solo quando viene combinata con chiavi a blocco hardware (Hardware Lock).

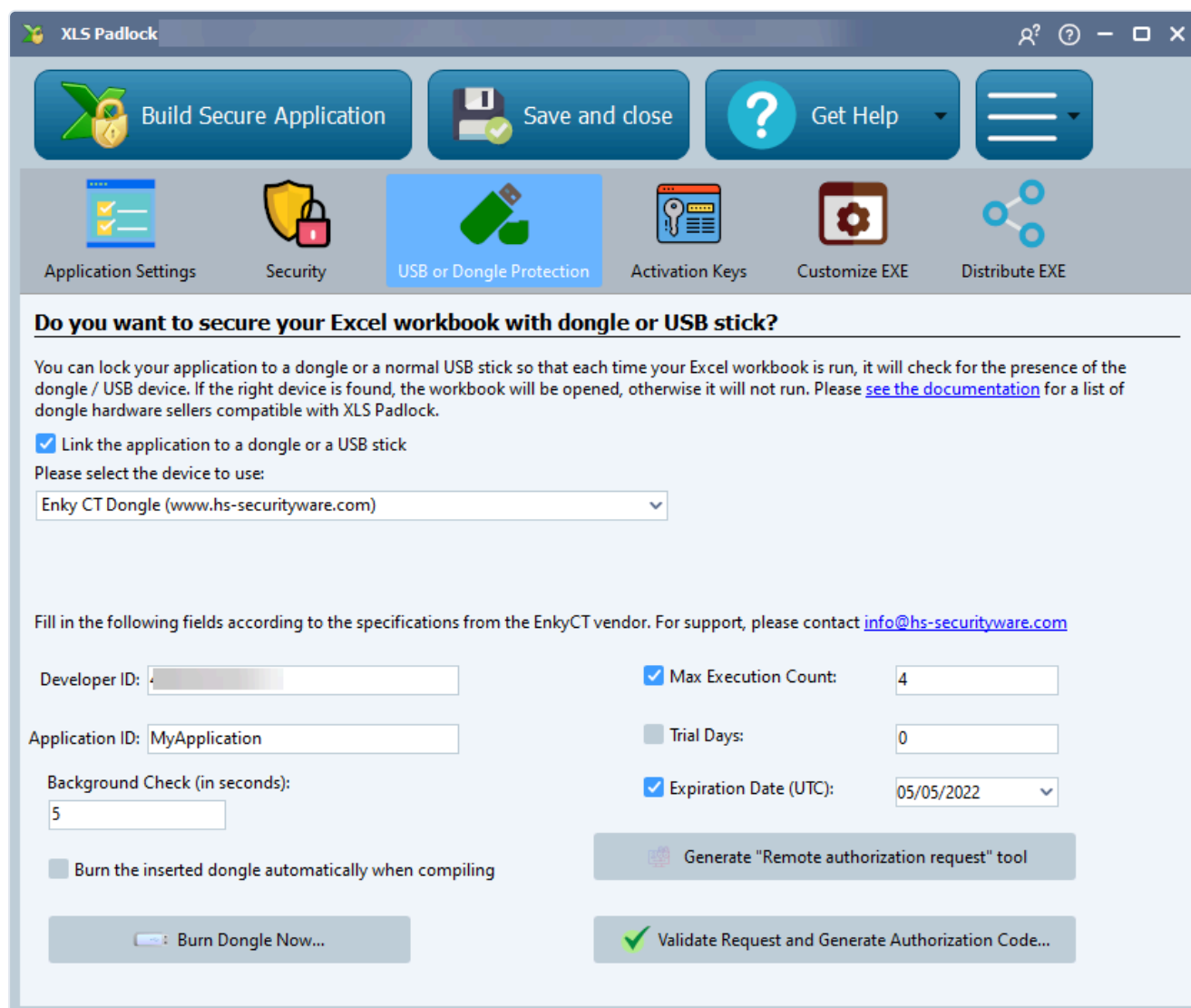
Quando il limite di una chiave (data, numero di esecuzioni o giorni di prova) viene raggiunto, all'utente verrà chiesto di inserire una nuova chiave per continuare a utilizzare l'applicazione:



Protezione tramite USB o dongle

Con XLS Padlock è possibile vincolare le cartelle di lavoro Excel a una chiavetta USB o a un dongle di sicurezza dedicato. Ciò significa che **l'applicazione della cartella di lavoro compilata non funzionerà se il dispositivo USB corretto non è inserito**. L'applicazione verifica la presenza del dispositivo all'avvio e può anche verificarla regolarmente durante l'esecuzione.

Per utilizzare questa protezione, selezionare l'opzione "**Link the application to a dongle or a USB stick**" (collega l'applicazione a un dongle o a una chiavetta USB) nella pagina "USB Protection" e selezionare il modello di dispositivo in proprio possesso.



The screenshot shows the XLS Padlock application window. The title bar reads "XLS Padlock". The main menu includes "Build Secure Application", "Save and close", "Get Help", and a hamburger menu. Below the menu is a navigation bar with icons for "Application Settings", "Security", "USB or Dongle Protection" (which is highlighted), "Activation Keys", "Customize EXE", and "Distribute EXE".

The "USB or Dongle Protection" section is titled "Do you want to secure your Excel workbook with dongle or USB stick?". It contains the following text: "You can lock your application to a dongle or a normal USB stick so that each time your Excel workbook is run, it will check for the presence of the dongle / USB device. If the right device is found, the workbook will be opened, otherwise it will not run. Please [see the documentation](#) for a list of dongle hardware sellers compatible with XLS Padlock."

There is a checked checkbox for "Link the application to a dongle or a USB stick". Below it, a dropdown menu is set to "Enky CT Dongle (www.hs-securityware.com)".

A note says: "Fill in the following fields according to the specifications from the EnkyCT vendor. For support, please contact info@hs-securityware.com".

Fields and options include:

- Developer ID: [text input]
- Application ID: MyApplication
- Background Check (in seconds): 5
- Max Execution Count: 4
- Trial Days: 0
- Expiration Date (UTC): 05/05/2022
- Burn the inserted dongle automatically when compiling
-
-
-

XLS Padlock supporta diversi tipi di protezione hardware per vincolare l'applicazione Excel a un dispositivo fisico.

Dongle Enky CT

Secondo il fornitore, Enky CT è un dongle con orologio temporale basato su smart card a 32 bit, semplice da usare e flessibile, utilizzato principalmente per la protezione del software e la limitazione temporale.

Per assistenza, contattare info@hs-securityware.com o visitare <https://hs-securityware.com>.

Come utilizzare il dongle Enky CT

Inserire il **Developer ID** fornito da HS Securityware (fa distinzione tra maiuscole e minuscole).

L'**Application ID** può essere qualsiasi valore desiderato per identificare il dongle e deve essere correlato all'applicazione che si sta creando.

Background Check

L'applicazione verifica la presenza del dongle all'avvio e può anche verificarla regolarmente durante l'esecuzione. È possibile definire la frequenza di queste verifiche; ad esempio, un valore di 20 indica che l'applicazione verificherà la presenza del dongle ogni 20 secondi. Se il dongle non viene trovato, compare una schermata di avviso (nag screen) che obbliga l'utente a inserire il dongle entro 15 secondi, altrimenti l'applicazione si chiuderà.

Set restrictions on the application

È possibile limitare il numero di volte in cui l'applicazione può essere eseguita impostando un **Max Execution Count**. È inoltre possibile far scadere l'applicazione dopo un determinato numero di giorni (**Trial Days**) o a una data specifica (**Expiration Date**).

Funzioni di aggiornamento remoto del dongle

Questo modello di dongle consente di aggiornare a distanza i dongle già in possesso dei clienti. Ad esempio, è possibile prolungare una data di scadenza o aggiungere altre esecuzioni. Questa funzione di aggiornamento remoto si basa su un sistema di codice di richiesta e di autorizzazione.

Come aggiornare a distanza il dongle di un cliente:

1. Innanzitutto, generare lo strumento "remote authorization request tool" da XLS Padlock e inviare al cliente i file EXE e DAT ottenuti.

Avvertenza

Questi due file (`.EXE` e `.DAT`) devono rimanere nella stessa cartella affinché lo strumento funzioni.

2. Il cliente esegue lo strumento per generare un **request code**, che invia all'utente.
3. In XLS Padlock, scegliere "Validate Request and Generate Authorization Code", incollare il request code e fare clic su "Validate Request Code". È quindi possibile visualizzare lo stato attuale del dongle e impostare nuovi limiti (numero di utilizzi o una nuova data di scadenza).
4. Generare l'**authorization code** e inviarlo nuovamente al cliente. Quest'ultimo utilizzerà lo stesso strumento per inserire il codice e applicare l'aggiornamento al proprio dongle.

Dongle Enky LC

Per assistenza, contattare info@hs-securityware.com o visitare <https://hs-securityware.com>.

L'Enky LC è un dongle HID economico e senza driver per la protezione del software.

Come utilizzare il dongle Enky LC2

1. Inserire il **Developer ID** ricevuto da HS-Security Ware.
2. Inserire un **Product ID** univoco per la propria cartella di lavoro. Ciò garantisce che vengano accettati solo i dongle con il Product ID corretto.

Passaggi per configurare un dongle Enky LC2

Per collegare un dongle all'applicazione, è necessario prima "masterizzarlo" tramite il pulsante "**Burn Dongle Now**" in XLS Padlock. Questa operazione configura il dongle con i propri identificativi. Questo processo può avvenire anche automaticamente durante la compilazione dell'applicazione se è collegato un dongle compatibile.

Background Check

È possibile attivare una verifica in background (background check) per garantire che il dongle rimanga collegato durante l'esecuzione dell'applicazione. È possibile impostare l'intervallo (in secondi) con cui viene eseguita la verifica.

Protezione tramite chiavetta USB generica

È possibile vincolare l'applicazione a una o più chiavette USB generiche in base al loro ID univoco del produttore.

- Innanzitutto, è necessario inserire un **Application Secret ID**. Questo viene combinato con l'ID della chiavetta USB per generare un hash univoco.
- All'avvio dell'applicazione, vengono analizzate tutte le unità USB. Se viene trovata un'unità con un hash autorizzato, la cartella di lavoro si apre.

Per autorizzare una chiavetta USB, collegarla, selezionarla dall'elenco delle unità in XLS Padlock e fare clic su "Allow this USB disk". È possibile autorizzare più chiavette USB.

NOTA

La protezione tramite chiavetta USB offre minore flessibilità e sicurezza rispetto alla protezione tramite dongle dedicato.

Disattivazione

Introduzione alla disattivazione

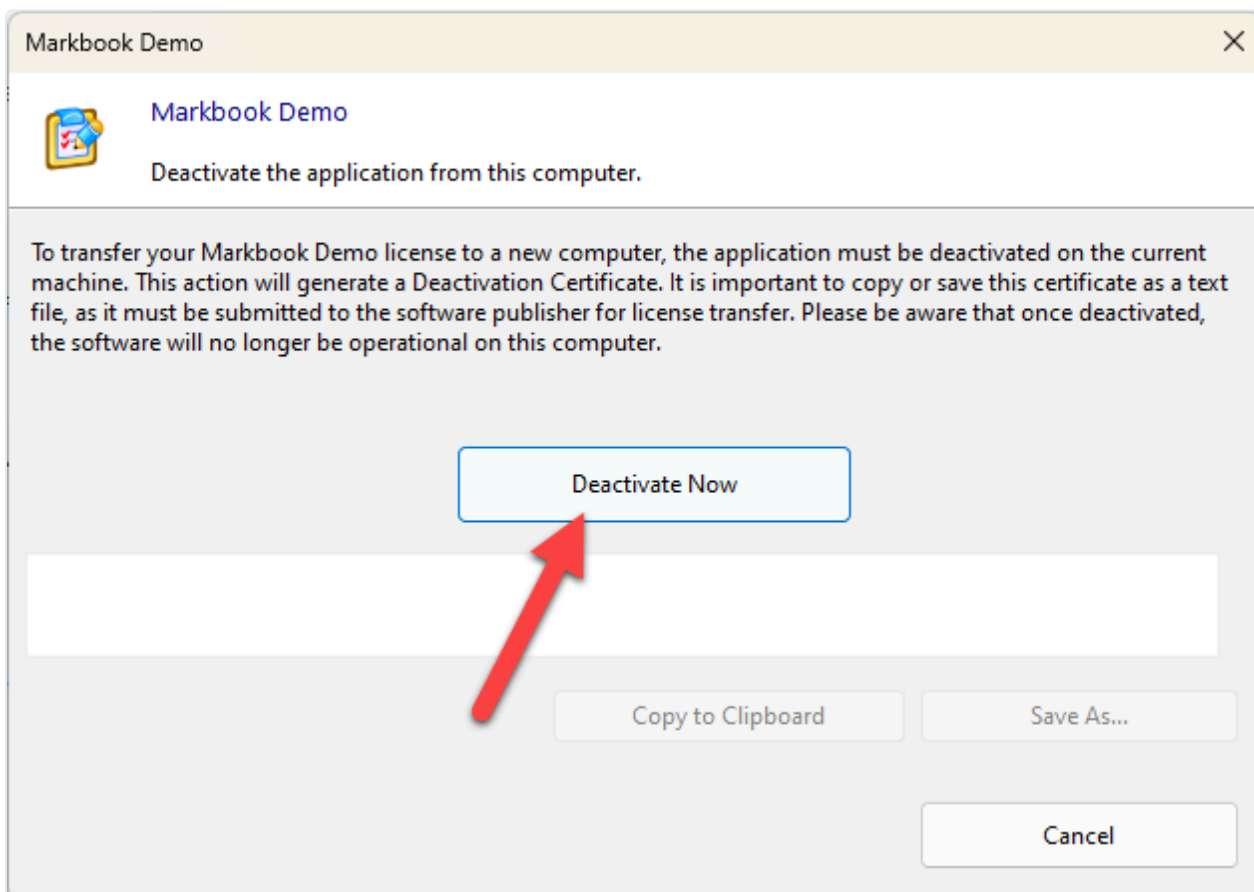
XLS Padlock offre un robusto sistema di disattivazione che consente agli utenti finali di annullare la registrazione della propria applicazione da un computer. Ciò è particolarmente utile quando i clienti devono trasferire una licenza su una nuova macchina o riattivare il proprio software dopo modifiche significative al sistema.

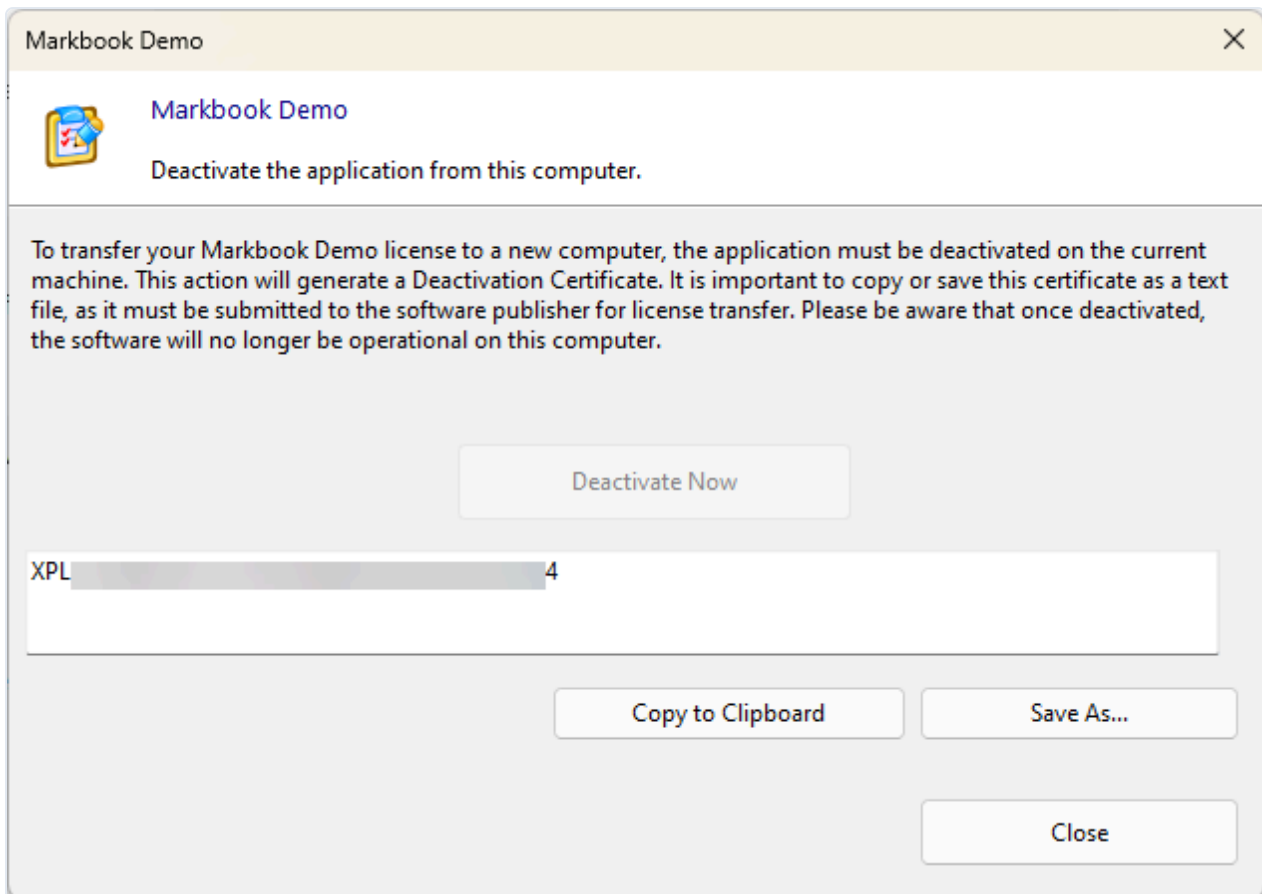
Come funziona la disattivazione

La disattivazione genera un certificato che l'utente finale deve inviare. Tramite la funzione **Test Deactivation Certificate** (verifica del certificato di disattivazione) di XLS Padlock, è possibile verificare l'autenticità del certificato. È fondamentale notare che, una volta disattivata un'applicazione, la sua Activation Key viene invalidata. Sarà necessario rilasciare una nuova chiave all'utente per qualsiasi attivazione futura.

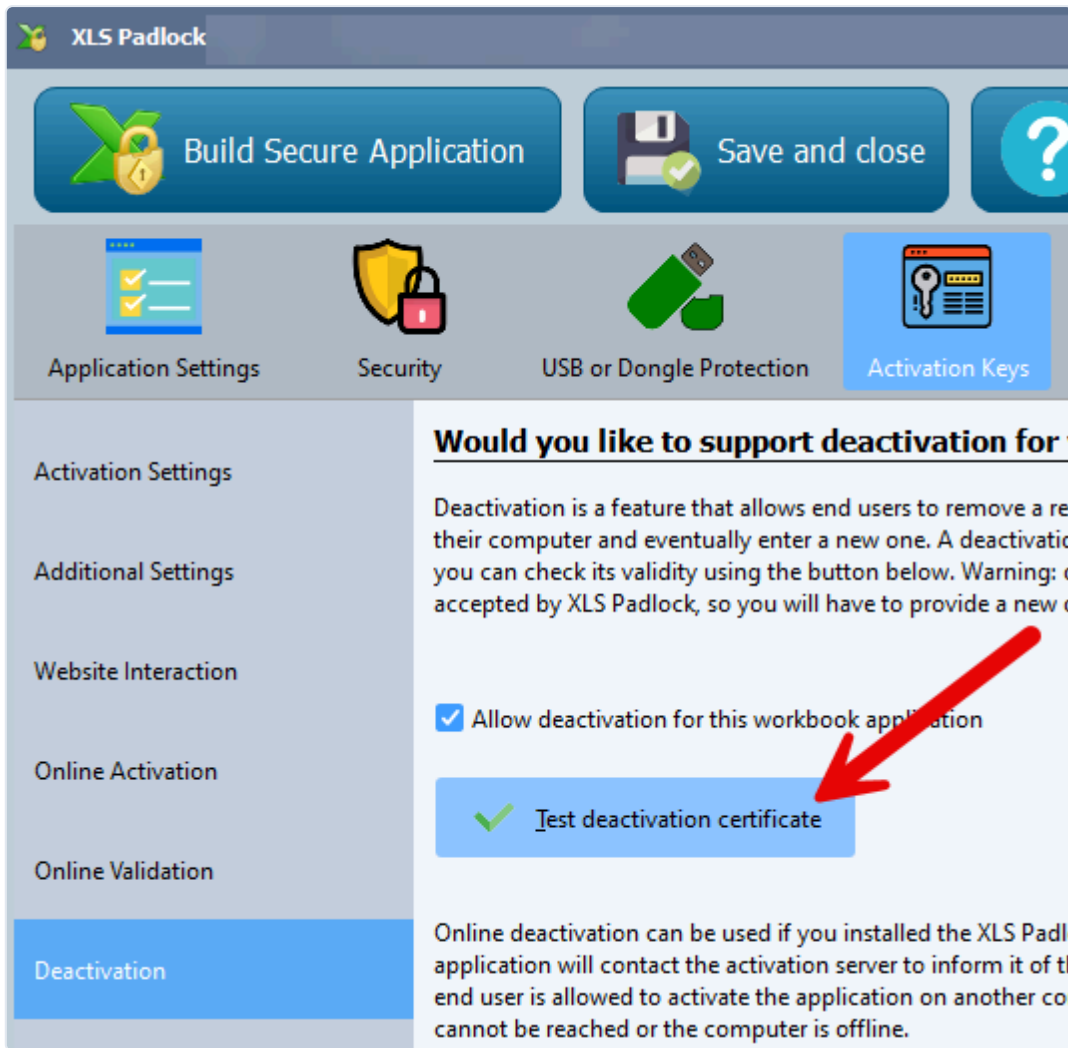
Disattivazione manuale

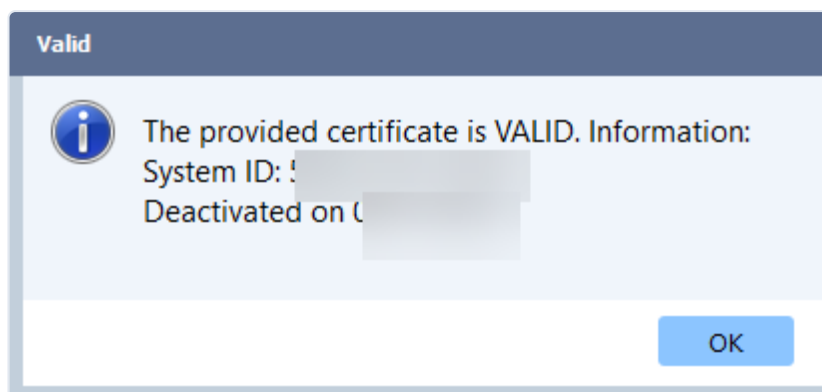
Quando un utente disattiva la propria applicazione, deve generare un certificato di disattivazione e inviarlo:





Alla ricezione, è possibile utilizzare il pulsante 'Test deactivation certificate' in XLS Padlock per verificare il certificato. Se la disattivazione va a buon fine, XLS Padlock visualizzerà la data di disattivazione e l'identificatore univoco del computer dell'utente:

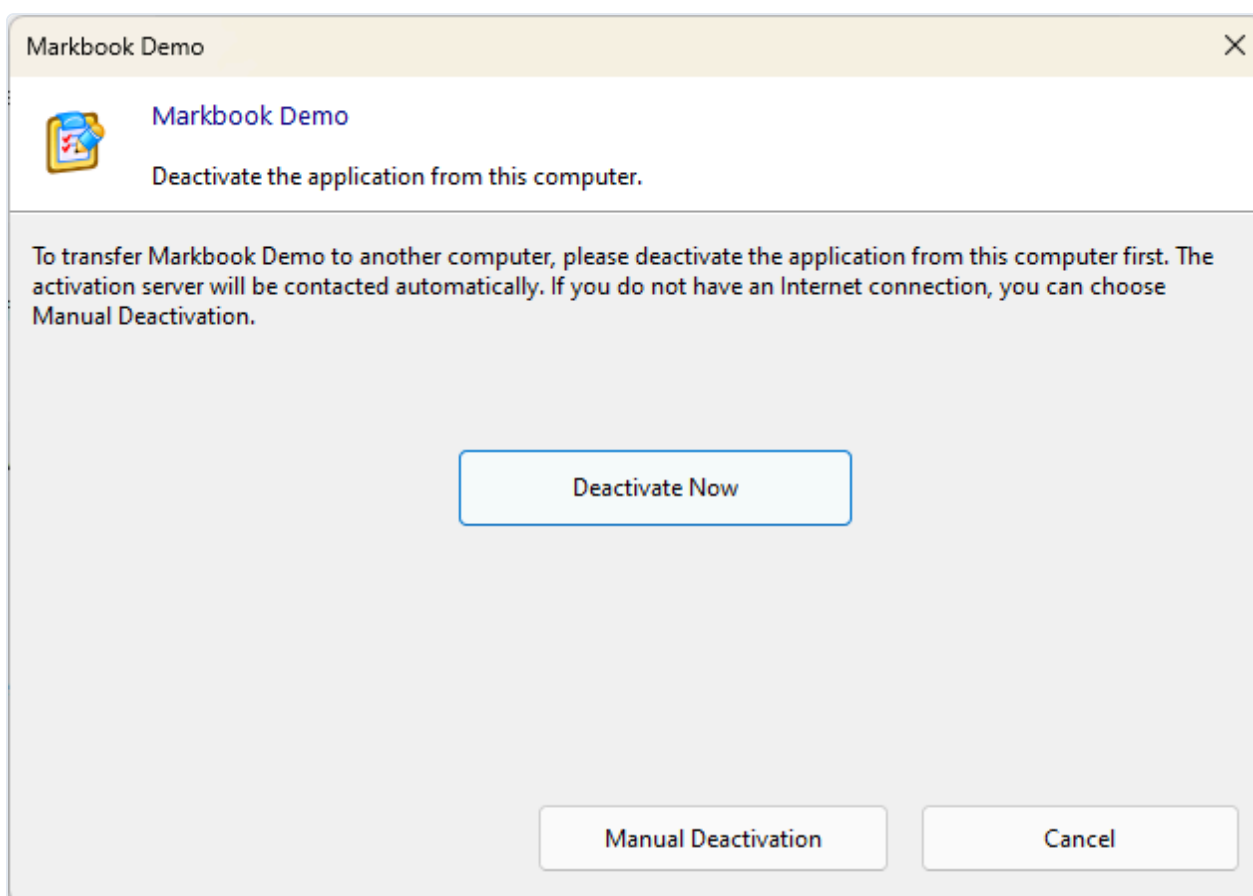




Queste informazioni Le consentono di gestire le nuove attivazioni per il Suo cliente.

Disattivazione online

Per un'esperienza più fluida, XLS Padlock supporta la disattivazione online se sul Suo server web è installato il XLS Padlock Activation Kit. L'applicazione comunica con il Suo server per annullare automaticamente la registrazione dell'attivazione. Se il server è irraggiungibile o il computer è offline, come soluzione di ripiego viene creato un certificato di disattivazione manuale.



La disattivazione online automatizza l'intero processo, rendendolo più semplice sia per Lei sia per i Suoi clienti.

Configurazione della disattivazione online

Base Deactivation URL

Per utilizzare la disattivazione online, è necessario fornire l'URL del XLS Padlock Activation Kit o del XLS Padlock WooCommerce Integration Kit sul Suo server web. Per esempio, se ha installato l'activation kit in una sottocartella denominata "activation", l'URL sarebbe

```
https://www.yourdomain.com/activation/dodeactivation .
```

Utilizzare HTTPS

Sono supportate le connessioni sicure tramite TLS/SSL. È consigliabile utilizzare sempre URL che iniziano con `https://`.

⚠ Se preferisce gestire la disattivazione manualmente, lasciare vuoto questo campo.

Hide Manual Deactivation Button (in tal caso viene mostrato se la disattivazione automatizzata non riesce)

Per impostazione predefinita, la disattivazione manuale è sempre disponibile come soluzione di ripiego. Se desidera incoraggiare gli utenti a utilizzare la disattivazione online, può scegliere di nascondere il pulsante di disattivazione manuale.

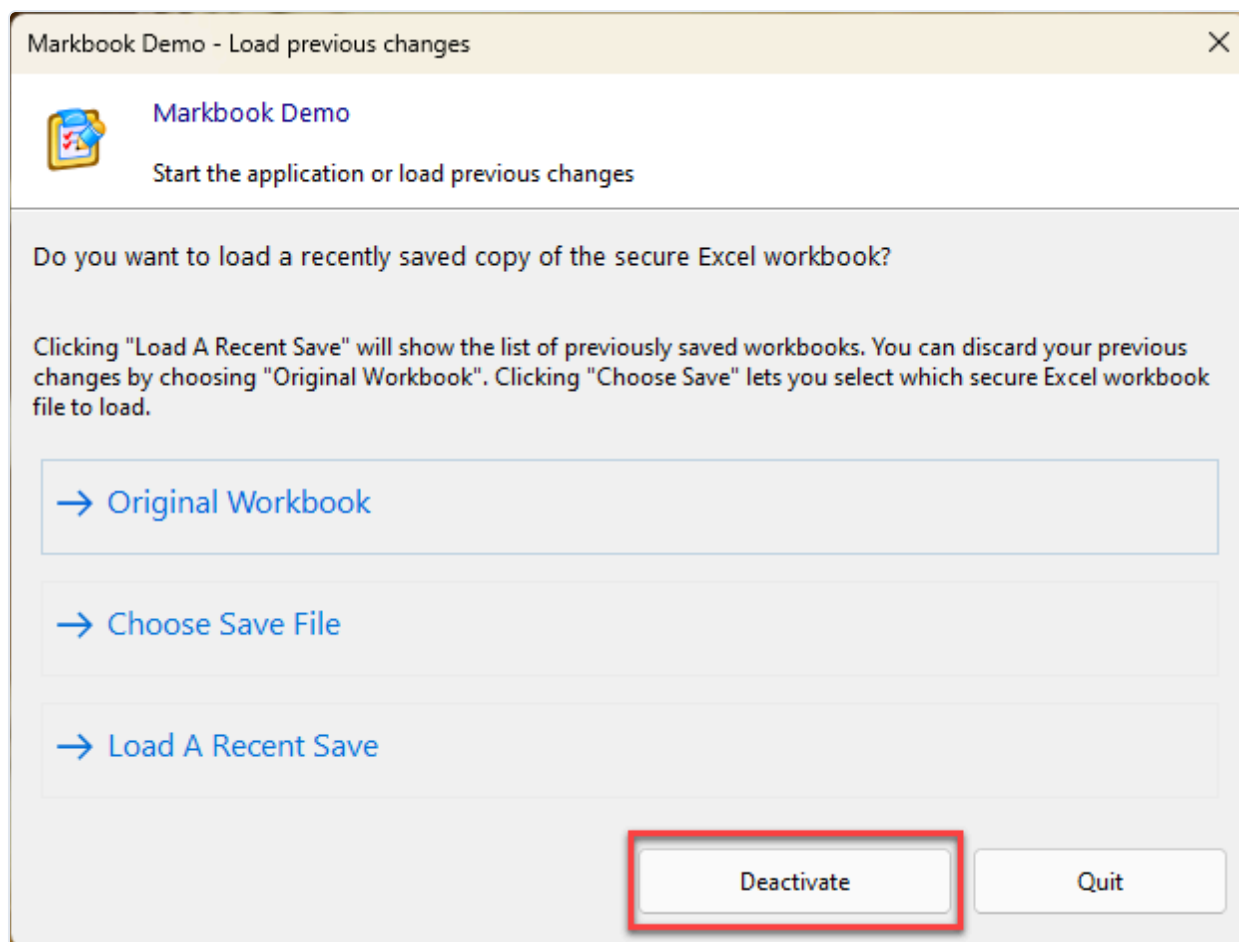
NOTA

Anche se il pulsante è nascosto, l'applicazione offrirà automaticamente la disattivazione manuale in caso di fallimento del processo online (per esempio, se il server di disattivazione è inaccessibile o se l'utente è offline).

Come avviare la disattivazione

Un utente può avviare il processo di disattivazione in tre modi:

- La prima opzione passa attraverso la finestra di dialogo Welcome integrata dell'applicazione. All'avvio dell'applicazione protetta, compare la finestra di benvenuto che presenta varie opzioni, tra cui un pulsante "Deactivate". Facendo clic su questo pulsante, gli utenti possono avviare il processo di disattivazione in modo intuitivo e guidato:



- Il secondo metodo ricorre alla riga di comando. Gli utenti avanzati possono preferire questo metodo per l'automazione o l'integrazione di sistema. Per farlo, l'utente deve aprire il prompt dei comandi di Windows, posizionarsi nella directory in cui si trova il file eseguibile dell'applicazione protetta e quindi digitare il nome dell'eseguibile seguito dal parametro `** -deact**`. Per esempio, se l'eseguibile si chiama `Application.exe`, il comando completo sarebbe:

```
C:\Path\to\application\Application.exe -deact
```

Premendo Invio, il parametro viene trasmesso al file eseguibile, avviando il processo di disattivazione.

- Il terzo metodo utilizza l'API VBA fornita da XLS Padlock.

La disattivazione è definitiva su una macchina

Si tenga presente che, una volta disattivata una chiave su un computer, essa viene definitivamente inserita nella lista nera su quella macchina e non può essere riutilizzata. La disattivazione è un'azione definitiva per quel sistema specifico. Per utilizzare nuovamente l'applicazione sullo stesso computer, sarà necessaria una nuova Activation Key. Si noti che ciò riguarda le Activation Keys, non i token di attivazione (come definiti nel WooCommerce Integration Kit).

👉 Vedere anche:., [Attivazione online](#chapter-online-activation), La finestra di dialogo Welcome integrata

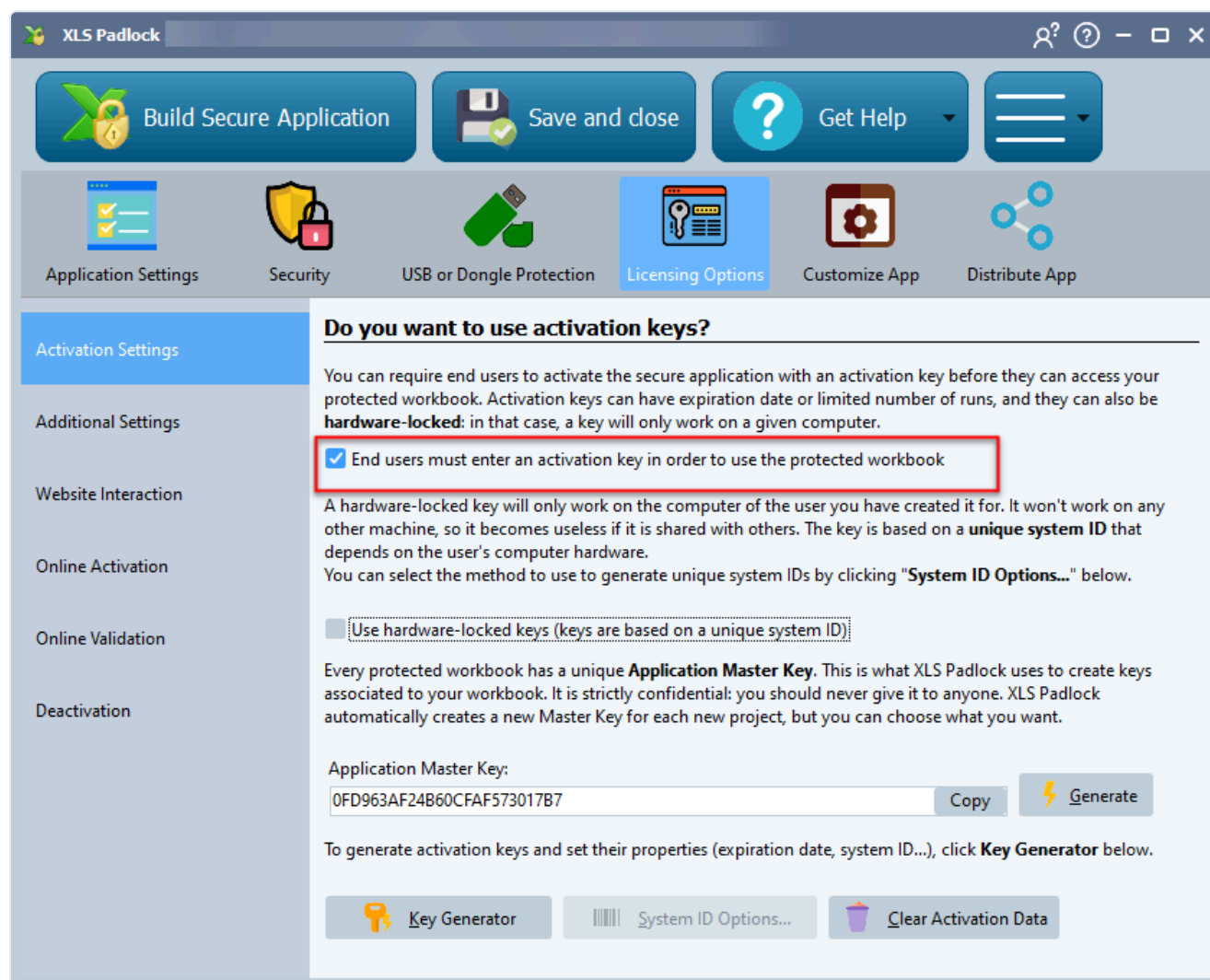
Creare cartelle di lavoro di prova

È possibile offrire un'anteprima delle proprie cartelle di lavoro Excel a potenziali clienti creando una versione di prova che richiede una licenza per sbloccare tutte le funzionalità. Questo è possibile grazie alle [chiavi di attivazione di XLS Padlock](#).

Questa guida mostra come configurare un periodo di prova per la propria applicazione basata su cartella di lavoro.

1. Abilitare le chiavi di attivazione

Per prima cosa, abilitare l'opzione "**End users must enter an activation key in order to use the protected workbook**" (gli utenti finali devono inserire una chiave di attivazione per utilizzare la cartella di lavoro protetta) e ricompilare l'applicazione.



The screenshot shows the XLS Padlock application window. The top bar contains buttons for 'Build Secure Application', 'Save and close', 'Get Help', and a menu icon. Below this is a navigation bar with icons for 'Application Settings', 'Security', 'USB or Dongle Protection', 'Licensing Options' (which is selected), 'Customize App', and 'Distribute App'. The main content area is titled 'Do you want to use activation keys?' and contains the following text:

You can require end users to activate the secure application with an activation key before they can access your protected workbook. Activation keys can have expiration date or limited number of runs, and they can also be **hardware-locked**: in that case, a key will only work on a given computer.

End users must enter an activation key in order to use the protected workbook

A hardware-locked key will only work on the computer of the user you have created it for. It won't work on any other machine, so it becomes useless if it is shared with others. The key is based on a **unique system ID** that depends on the user's computer hardware.

You can select the method to use to generate unique system IDs by clicking "**System ID Options...**" below.

Use hardware-locked keys (keys are based on a unique system ID)

Every protected workbook has a unique **Application Master Key**. This is what XLS Padlock uses to create keys associated to your workbook. It is strictly confidential: you should never give it to anyone. XLS Padlock automatically creates a new Master Key for each new project, but you can choose what you want.

Application Master Key:
0FD963AF24B60CFAF573017B7 Copy Generate

To generate activation keys and set their properties (expiration date, system ID...), click **Key Generator** below.

Key Generator System ID Options... Clear Activation Data

2. Generare una chiave di attivazione di prova

Aprire il [Key Generator](#) in XLS Padlock.

Per definire un periodo di prova, è possibile impostare un limite **Trial days** (giorni di prova, ad esempio 15 giorni), una **Expiration Date** (data di scadenza) fissa oppure un **Max Execution Count** (numero massimo di esecuzioni).

XLS Padlock - Application Key Generator

Please fill in the following fields in order to generate a **new activation key** to send to the end user who wants to run your protected workbook. Click **Generate** to output an activation key.
If you want to **generate several keys in mass**, [use our stand-alone key generator](#) (available for registered users only).

System ID provided by the end user:
8FHA-4DEB-B775 Paste From Clipboard

Max Execution Count: 0

Trial Days: 30

Expiration Date (UTC): 29/11/2020

Display nag screen (useful for trials)

Do not perform online validation for this key

Output Activation Key:
3F82H-8DFF9-85164-B6216-BAC1H-DC52B

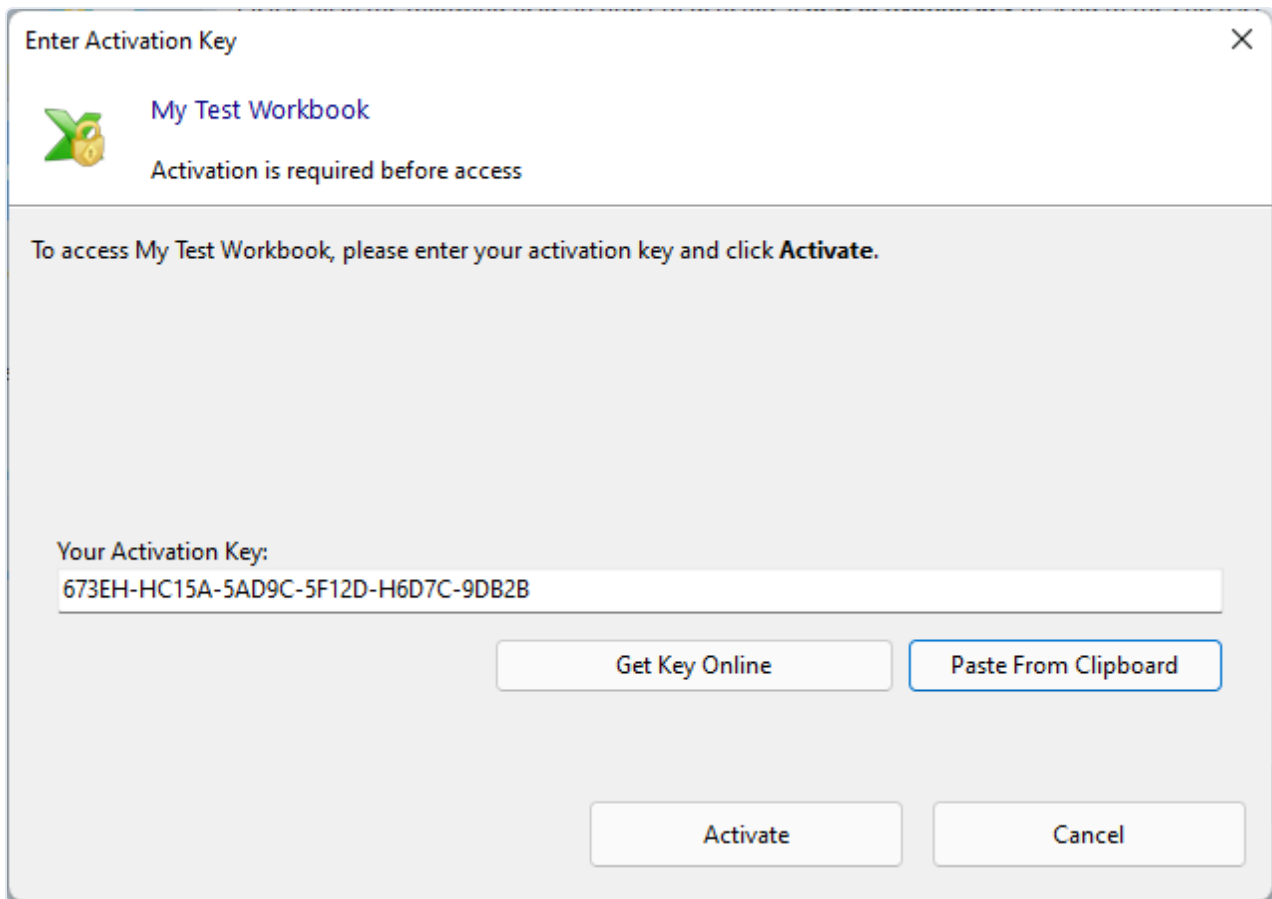
⚡ Generate 📄 Copy to clipboard 💾 Save As... ✅ Close

Per ricordare ai clienti che stanno utilizzando una versione di prova, selezionare **"Display nag screen"** (mostra schermata di promemoria). Questo mostrerà una finestra di dialogo di promemoria a ogni avvio dell'applicazione.

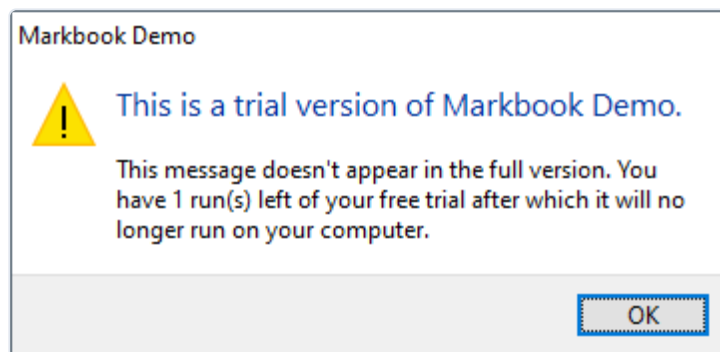
Fare clic su **Generate** per creare la chiave di prova, che potrà poi essere distribuita ai propri utenti.

Come funziona l'applicazione in modalità di prova

I clienti inseriranno la chiave di prova per attivare l'applicazione.



Una volta attivata, la cartella di lavoro si aprirà, ma comparirà la schermata di promemoria per indicare la durata di prova rimanente.



Al termine del periodo di prova, l'applicazione richiederà una nuova chiave di attivazione non di prova per continuare a funzionare.

Enter Activation Key [X]

Your activation key has expired. Please provide a new one.
To access Markbook Demo, please enter your activation key and click **Activate**.
You will have to provide the following system ID in order to receive an activation key:

System ID:

Your Activation Key:

Verificare lo stato di prova con VBA

È anche possibile [verificare a livello di codice se la cartella di lavoro compilata si trova in stato di prova](#) utilizzando l'API VBA.

Chiudere dopo un tempo prestabilito

Come posso impedire a un utente di lasciare una versione di prova aperta indefinitamente?

Un utente potrebbe ottenere una chiave di prova di 1 giorno, aprire il file EXE e lasciarlo aperto per sempre. L'applicazione continuerà a funzionare. Come si può impedirlo?

****Soluzione****: è possibile utilizzare VBA per impostare un timer con `Application.OnTime` che chiude automaticamente la cartella di lavoro dopo una durata specificata. Questo è un modo efficace per far rispettare i limiti della versione di prova.

Combinare questo con le opzioni di sicurezza di XLS Padlock per vietare l'accesso all'editor VBA e compilare il codice VBA, in modo da impedire agli utenti di disattivare il timer.

Implementazione

Inserire il codice seguente nel modulo `ThisWorkbook` del progetto VBA.

```
' --- In ThisWorkbook module ---

' Variable to store the scheduled time for the timer
Private mScheduledTime As Date

' This procedure will be called by the timer to close the workbook
Public Sub CloseAndSave()
    ' Save any changes and close the workbook
    ThisWorkbook.Close SaveChanges:=True
End Sub

' This event runs when the workbook is opened, starting the timer
Private Sub Workbook_Open()
    ' Set the timer to run the "CloseAndSave" procedure in 8 hours.
    ' You can change the time value as needed.
    mScheduledTime = Now + TimeValue("08:00:00")
    Application.OnTime EarliestTime:=mScheduledTime, Procedure:="ThisWorkbook.CloseAndSave"

    ' Optional: Inform the user that the application will close automatically.
    ' MsgBox "This application will automatically close in 8 hours.", vbInformation
End Sub

' This event runs just before the workbook closes
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    ' Cancel the scheduled OnTime event to prevent errors if the user
    ' closes the workbook manually before the timer runs.
    On Error Resume Next
    Application.OnTime EarliestTime:=mScheduledTime, Procedure:="ThisWorkbook.CloseAndSave", Sc
End Sub
```

Come funziona

1. **Workbook_Open** : all'avvio dell'applicazione, questo evento pianifica l'esecuzione della macro `CloseAndSave` dopo 8 ore (`TimeValue("08:00:00")`). È possibile regolare questa durata in base alle proprie esigenze (ad esempio, `TimeValue("01:00:00")` per un'ora).
2. **CloseAndSave** : è la macro che esegue l'azione. Salva la cartella di lavoro e poi la chiude.
3. **Workbook_BeforeClose** : è un passaggio di pulizia essenziale. Se l'utente chiude la cartella di lavoro manualmente, questo codice annulla il timer in attesa, impedendo a Excel di tentare di eseguire una macro su una cartella di lavoro che non è più aperta, cosa che provocherebbe un errore.

Verificare lo stato di prova

XLS Padlock fornisce un'API VBA che consente di determinare a livello di codice se la cartella di lavoro compilata è in esecuzione in stato di prova. Utilizzando la funzione seguente, è possibile abilitare o disabilitare funzionalità specifiche, visualizzare messaggi di aggiornamento o creare altre limitazioni di prova personalizzate in base allo stato della licenza dell'utente...

👉 Inserire la seguente funzione in un modulo VBA:

```
Public Function IsTrial()  
    Dim XLSPadlock As Object  
    On Error GoTo Err  
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormuLa").Object  
    IsTrial = XLSPadlock.PLEvalVar("IsTrial")  
    Exit Function  
Err:  
    IsTrial = False  
End Function
```

È quindi possibile chiamare la funzione:

```
Sub Test_IsTrial()  
    If IsTrial() Then  
        MsgBox "Trial"  
    Else  
        MsgBox "Registered"  
    End If  
End Sub
```

Avvertenza

La funzione restituirà true solo se la chiave di attivazione possiede il flag "display nag screen" (prova):

XLS Padlock - Application Key Generator

Please fill in the following fields in order to generate a **new activation key** to send to the end user who wants to run your protected workbook. Click **Generate** to output an activation key.
If you want to **generate several keys in mass**, [use our stand-alone key generator](#) (available for registered users only).

System ID provided by the end user:

Max Execution Count:

Trial Days:

Expiration Date (UTC):

Display nag screen (useful for trials)

Do not perform online validation for this key

Output Activation Key:

👉 Vedere anche: [Impostare restrizioni sulle chiavi di attivazione](#)

Giorni di prova rimanenti

Questa funzione VBA consente di **recuperare in modo programmatico il numero di giorni o di esecuzioni rimanenti** per una [cartella di lavoro in versione di prova](#). Funziona anche per le chiavi registrate che hanno una data di scadenza o un numero limitato di esecuzioni.

👉 Inserire la seguente funzione in un modulo VBA:

```
Public Function ReadTrialState()  
    Dim XLSPadlock As Object  
    On Error GoTo Err  
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object  
    ReadTrialState = XLSPadlock.PLEvalVar("TrialState")  
    Exit Function  
Err:  
    ReadTrialState = ""  
End Function
```

È quindi possibile richiamare la funzione:

```
Sub Test_Trial()  
    rdays = ReadTrialState()  
    Worksheets("Sheet1").Range("A1").Value = rdays  
End Sub
```

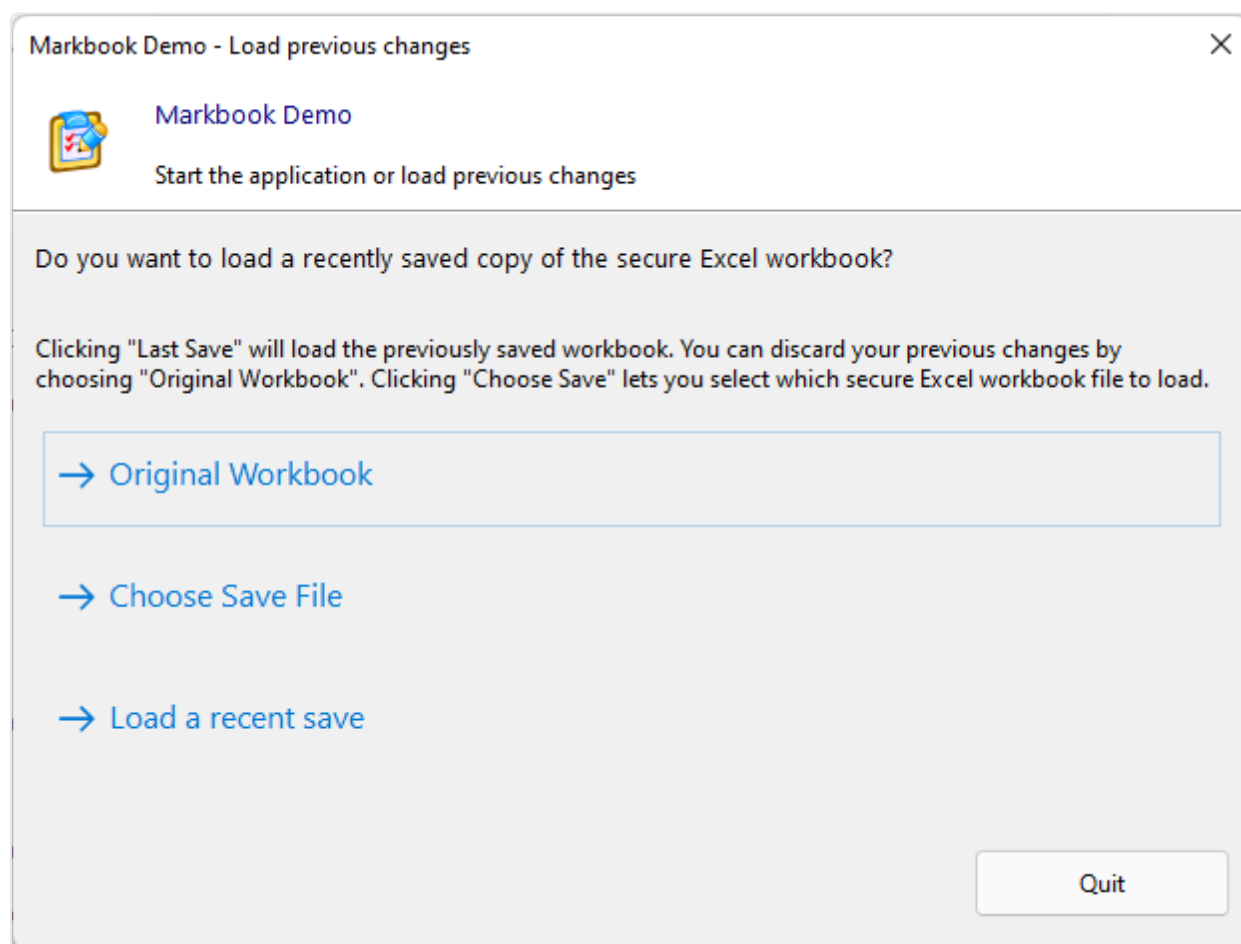
👉 Vedere anche: [Come creare cartelle di lavoro in versione di prova](#)

Opzioni di salvataggio della cartella di lavoro

Queste opzioni permettono di affinare il modo in cui le modifiche alla cartella di lavoro vengono salvate e caricate quando si [abilita il salvataggio](#).

Always Show the "Load Previous Changes" Welcome Screen

Per impostazione predefinita, se non esiste alcun file di salvataggio, l'applicazione apre direttamente la cartella di lavoro originale. Quando questa opzione è attivata, la finestra di dialogo di benvenuto viene sempre mostrata per prima, consentendo agli utenti di iniziare con la cartella di lavoro originale oppure di caricare un file di salvataggio esistente.



Ciò è utile se distribuite file di salvataggio predefiniti con la vostra applicazione oppure se spostate l'EXE su un altro computer e desiderate caricare un file di salvataggio esistente.

Save Changes Automatically and Load Them Without Prompt

Questa opzione fornisce un meccanismo di salvataggio automatico e trasparente. Le modifiche dell'utente vengono salvate in un file predefinito quando l'applicazione viene chiusa e vengono ricaricate

automaticamente all'avvio successivo. Non viene visualizzata alcuna finestra di dialogo "Save As" né alcuna schermata di benvenuto.

Con questa modalità, gli utenti non possono creare più file di salvataggio diversi.

Save Files Can Only Be Opened on the Computer They Were Saved On (Hardware-Locking)

Questa opzione aumenta la sicurezza bloccando i file di salvataggio sul computer specifico in cui sono stati creati. Un file di salvataggio creato su un PC non può essere aperto su un altro, impedendo agli utenti di condividere i propri dati salvati.

➔ [Scoprite di più sul blocco hardware dei file di salvataggio.](#)

Allow Save but Do Not Handle Loading/Saving

Attivate questa opzione se desiderate disabilitare le finestre di dialogo di salvataggio integrate di XLS Padlock e gestire personalmente l'intero processo di salvataggio e caricamento tramite codice VBA. Ciò vi conferisce il controllo completo, ad esempio per salvare una copia sicura senza alcuna richiesta all'utente.

➔ [Scoprite come salvare una copia sicura della cartella di lavoro senza richiesta tramite VBA.](#)

Do Not Display the "Original Workbook" Choice

Questa opzione rimuove la scelta "Start with the original workbook" dalla schermata di benvenuto, obbligando gli utenti a caricare un file `.XLSC` o `.XLSCE` salvato. Ciò è utile se la vostra applicazione è sempre destinata a essere utilizzata con file di dati specifici che fornite voi.

Rete di sicurezza automatica

Se il file di salvataggio precedente di un utente non riesce a caricarsi, ad esempio dopo che avete modificato la [Secret Key](#) per una nuova versione, oppure dopo aver cambiato modalità di salvataggio tra le versioni, il pulsante "Original Workbook" viene riabilitato automaticamente nella schermata di benvenuto per quel nuovo tentativo. Ciò garantisce che l'utente possa sempre recuperare l'applicazione invece di restare intrappolato in un ciclo di messaggi di errore.

👉 Vedere anche

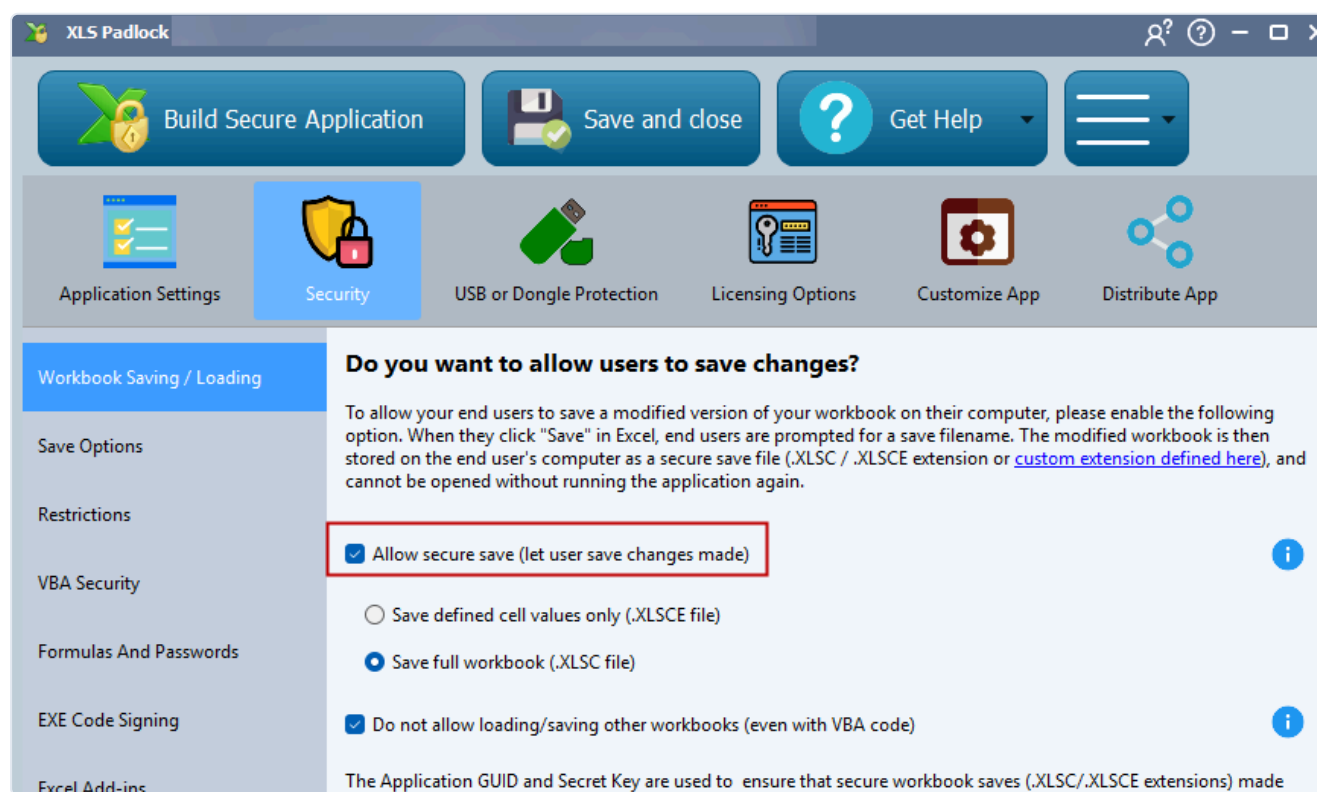
- [Panoramica del salvataggio e del caricamento delle cartelle di lavoro](#)
- [Modalità di salvataggio: cartella di lavoro completa o valori delle celle](#)
- [Ricette VBA per il salvataggio delle cartelle di lavoro](#)

Salvataggio e caricamento

Con XLS Padlock, avete il controllo completo sulla possibilità per gli utenti di salvare le modifiche alla vostra cartella di lavoro Excel.

Come XLS Padlock salva e carica le modifiche degli utenti

Per attivare o disattivare il salvataggio, selezionate l'opzione "**Allow secure save**" (Consenti salvataggio sicuro) nella pagina Security:

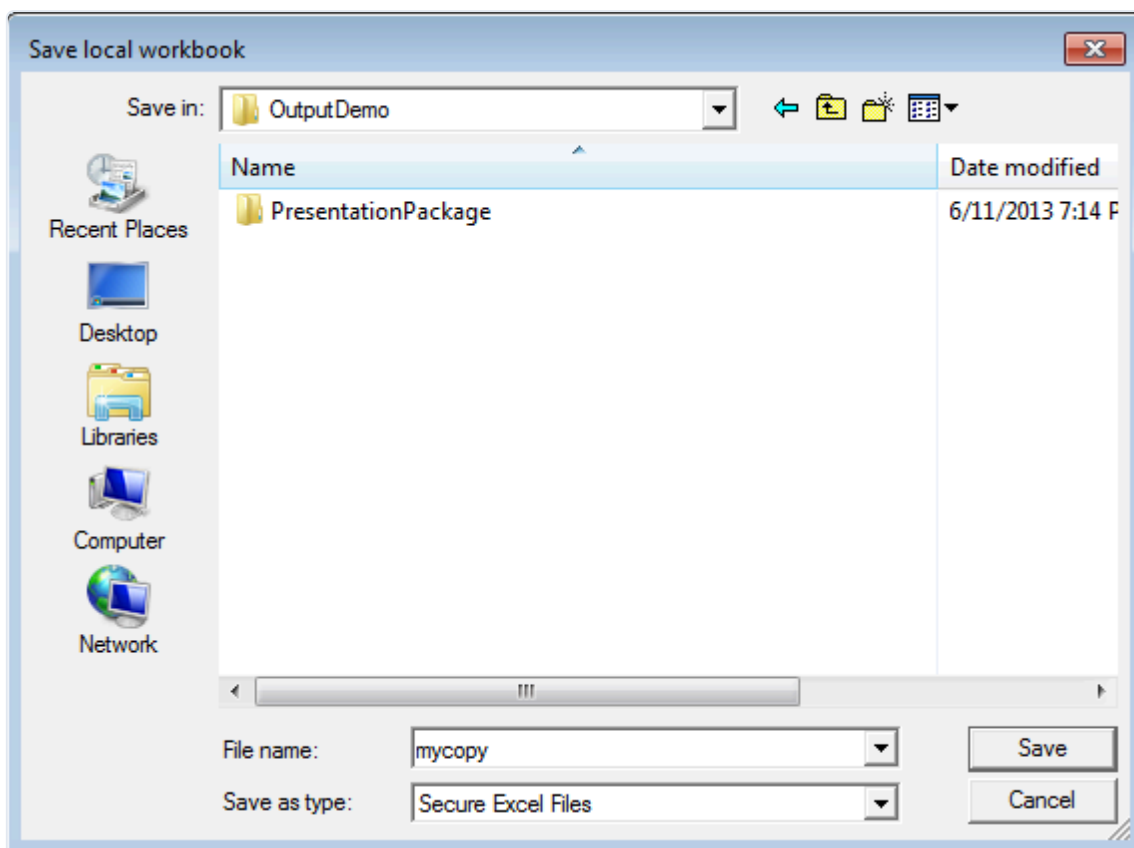


- **Se questa opzione è disattivata**, la vostra applicazione non verrà mai modificata. Qualsiasi modifica apportata da un utente viene scartata alla chiusura dell'applicazione. Notate che in alcune versioni di Excel il pulsante Salva non è disabilitato e sembrerà funzionare, ma in realtà nessuna modifica viene salvata.
- **Se questa opzione è attivata**, i vostri clienti potranno salvare le proprie modifiche. Dovrete quindi [selezionare una modalità di salvataggio](#).

Per salvare le modifiche, gli utenti finali possono fare clic sul pulsante Salva standard



in Excel oppure utilizzare il menu "File => Salva". Apparirà quindi la finestra di dialogo "Salva con nome", che chiede dove salvare la loro copia sicura della cartella di lavoro:



A questo file di salvataggio viene assegnata l'estensione `.XLSC` (o `.XLSCE`) e **non può essere aperto senza eseguire nuovamente l'applicazione sicura.**

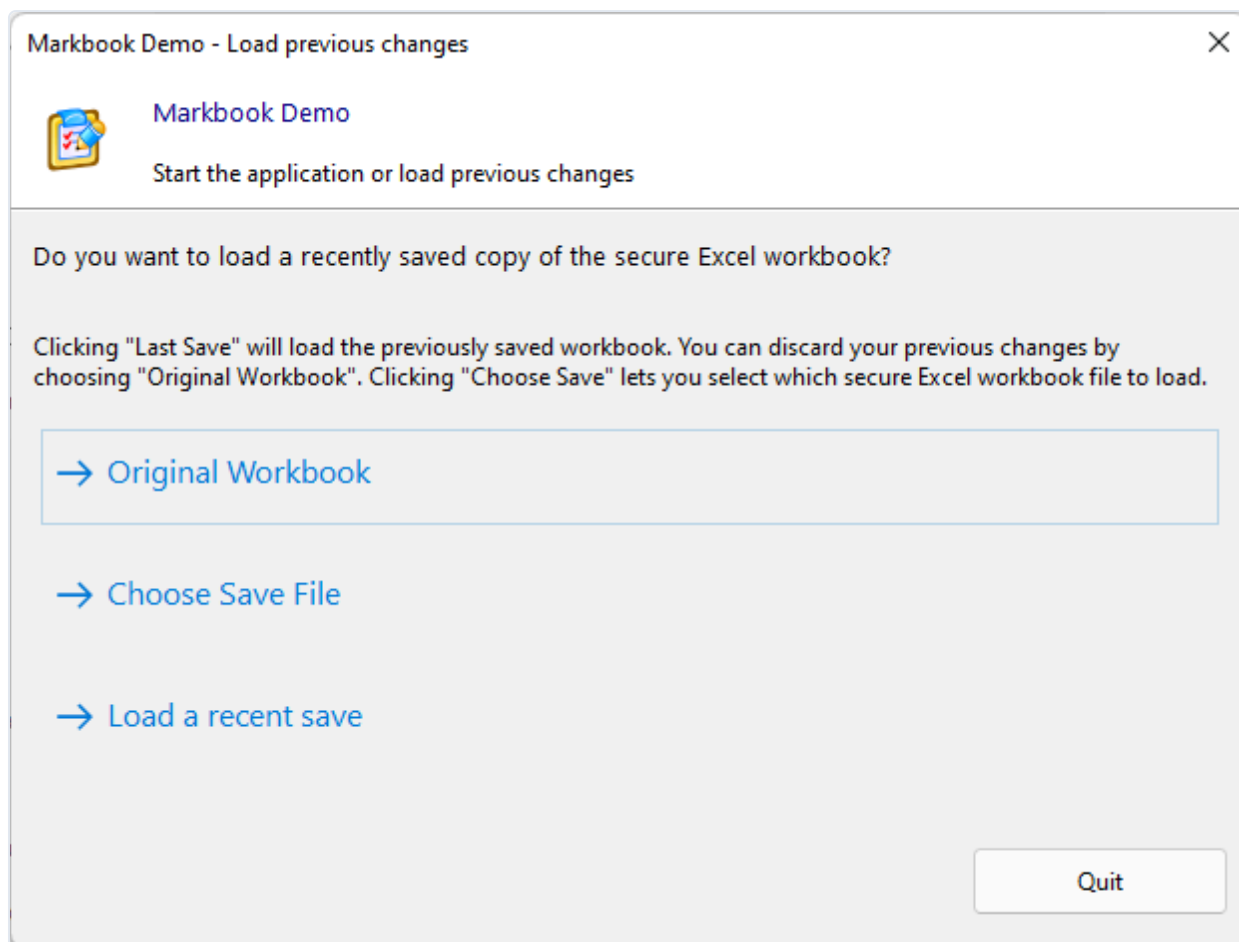
Per impostazione predefinita, la finestra di dialogo chiederà agli utenti finali se desiderano sovrascrivere le proprie modifiche. È possibile disattivare questo comportamento nelle [Opzioni avanzate](#).

Estensione personalizzata per i file di salvataggio

È possibile definire un'estensione personalizzata per i file di salvataggio nelle [Opzioni avanzate](#).

Come gli utenti finali caricano le proprie modifiche

La volta successiva che eseguiranno la vostra applicazione, i clienti avranno la possibilità di aprire la vostra cartella di lavoro originale oppure una versione precedentemente salvata. La schermata di benvenuto consente loro di "Choose Save File" (scegliere il file di salvataggio) o di caricare un file dal menu "Recent Saves" (salvataggi recenti).



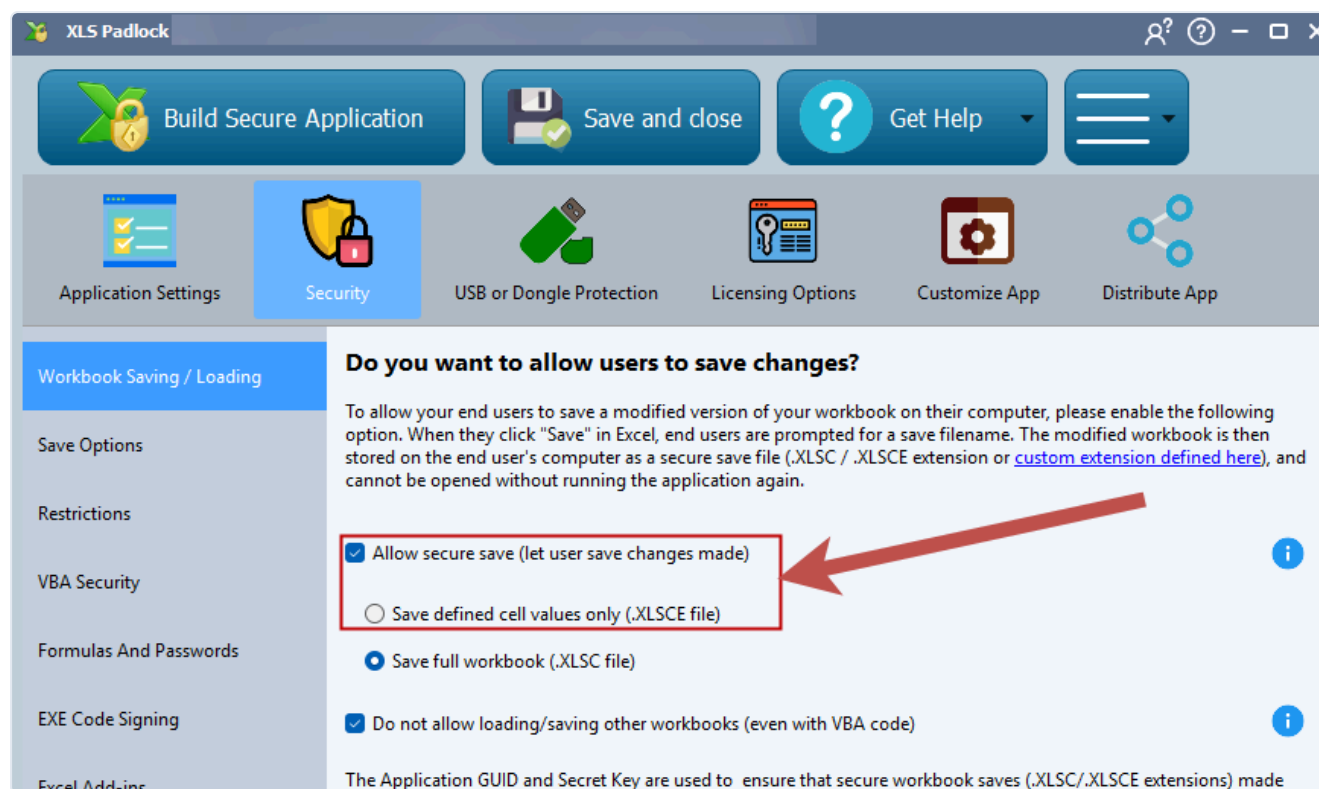
Per impostazione predefinita, la finestra di dialogo "Salva con nome" viene visualizzata ogni volta che un utente fa clic sul pulsante Salva. L'ultimo file salvato viene memorizzato automaticamente. Notate che funziona solo il comando `Save`; `Save As` è sempre disabilitato.

Vedere anche

- [Come migrare i dati utente da una versione precedente con VBA](#)
- [Scegliere come i dati vengono salvati nella cartella di lavoro](#)
- [Salvare e ripristinare per programmazione valori personalizzati con codice VBA](#)

Modalità di salvataggio: completo o valori delle celle

Se si consente agli utenti di salvare il proprio lavoro, è necessario definire come XLS Padlock salverà e caricherà tali modifiche. XLS Padlock offre due modalità di salvataggio, selezionabili nella pagina [Workbook Saving and Loading](#).



Salvare la cartella di lavoro completa (file .XLSC)

In questa modalità, una copia completa e cifrata della cartella di lavoro, comprensiva di tutte le modifiche dell'utente, viene memorizzata in un file sicuro `.XLSC`. Questo file può essere aperto solo dall'applicazione protetta, garantendo che la cartella di lavoro rimanga sicura.

La **modalità Full Save** (salvataggio completo) cifra e salva l'intera cartella di lavoro esattamente com'è al momento del salvataggio. Ciò significa che, se in seguito si distribuisce un EXE aggiornato, gli utenti che aprono un vecchio file di salvataggio vedranno il lavoro salvato in precedenza, e non i nuovi aggiornamenti. Il loro file di salvataggio è un'istantanea completa della cartella di lavoro in quel momento.

I file di salvataggio completo possono essere condivisi tra utenti, a meno che non li si [blocchi su una macchina specifica](#). È inoltre possibile [decifrare autonomamente questi file di salvataggio](#) per recuperare i dati degli utenti.

Quando utilizzare la modalità Full Save

La modalità Full Save è la scelta predefinita per la maggior parte degli utenti ed è particolarmente consigliata per cartelle di lavoro complesse in cui gli utenti apportano modifiche estese su più fogli.

Il principale svantaggio è che i file di salvataggio degli utenti non sono automaticamente compatibili con le versioni aggiornate dell'applicazione. Se si pubblicano frequentemente aggiornamenti alla logica o alla struttura della cartella di lavoro, conviene piuttosto valutare la modalità Cell Values (valori delle celle).

👉 Vedi anche: [Come migrare i dati utente da una versione precedente con VBA](#)

Salvare solo i valori delle celle definite (file .XLSCE)

In questa modalità, vengono salvati e ripristinati solo i valori di celle specifiche e predefinite. È la scelta ideale se si aggiorna frequentemente la cartella di lavoro di origine e si distribuiscono nuove versioni.

Quando un utente carica il proprio file di salvataggio `.XLSCE` con il nuovo EXE, i dati salvati vengono caricati nella struttura aggiornata della cartella di lavoro. Ciò gli consente di beneficiare degli aggiornamenti senza perdere il proprio lavoro.

Il principale svantaggio è che è necessario [definire manualmente quali celle salvare](#) prima di compilare l'applicazione.

Come i file `.XLSC`, anche i file `.XLSCE` sono cifrati in modo sicuro e possono essere aperti solo dalla propria applicazione. Ciò è garantito da una [chiave segreta univoca](#) specifica del proprio progetto.

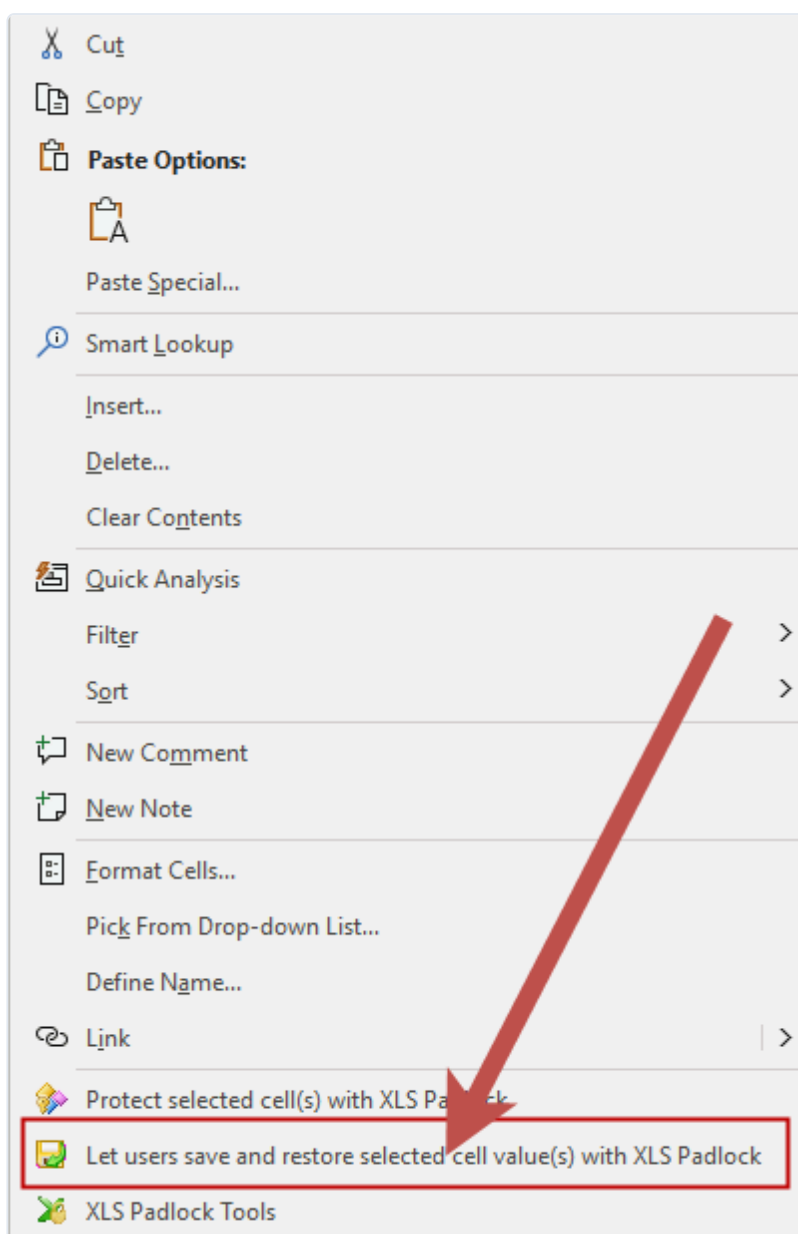
Scegliere la modalità di salvataggio fin dall'inizio

I formati Full Save (`.XLSC`) e Cell Values (`.XLSCE`) non sono intercambiabili. Una volta distribuita un'applicazione che utilizza una modalità, passare all'altra in una versione successiva significa che i file di salvataggio esistenti degli utenti non verranno caricati: per quegli utenti, al primo avvio della nuova versione, XLS Padlock ripristinerà silenziosamente la cartella di lavoro originale incorporata, e tutti i dati salvati nella modalità precedente andranno persi. Scegliere fin dall'inizio del progetto la modalità adatta alla propria strategia di aggiornamento e mantenerla.

Definire le celle da salvare e ripristinare

Quando si utilizza la modalità **Save defined cell values only**, è necessario specificare quali celle l'applicazione salverà e ripristinerà.

👉 Per farlo, è sufficiente **fare clic con il pulsante destro del mouse su una o più celle** e scegliere **"Let users save and restore selected cell value(s) with XLS Padlock"** dal menu contestuale.

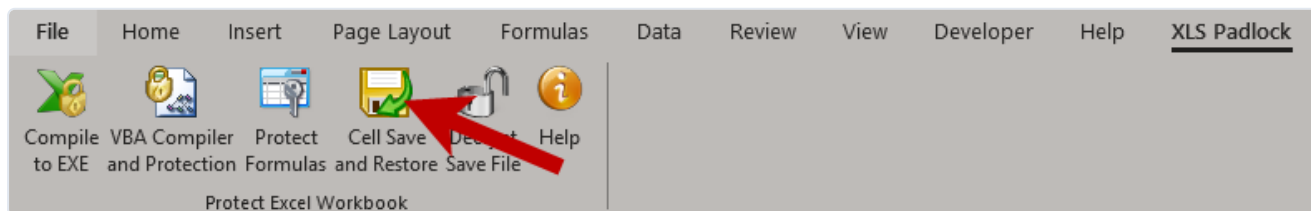


XLS Padlock confermerà che le celle selezionate sono contrassegnate per il salvataggio e il ripristino.

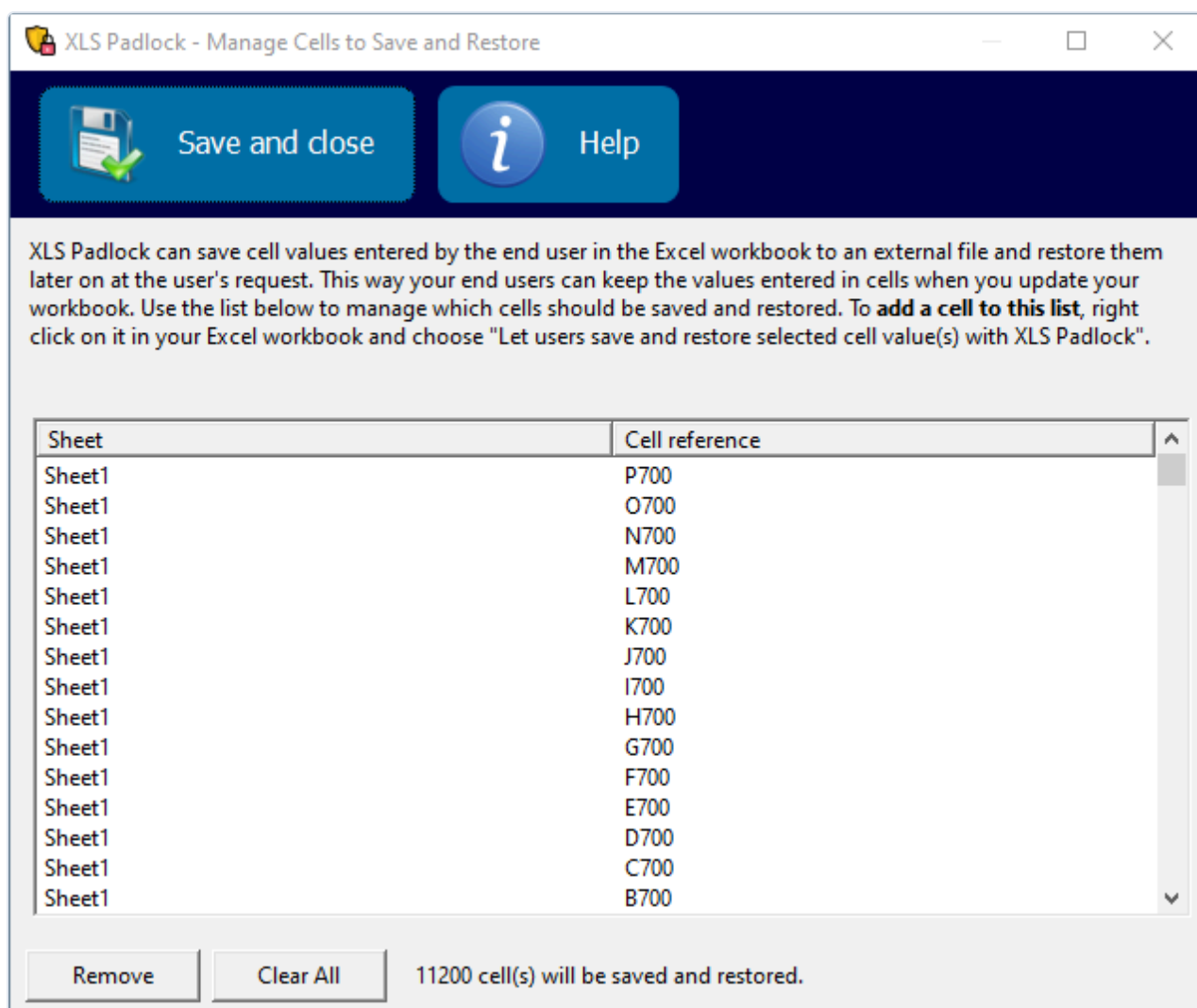
È anche possibile [memorizzare e ripristinare valori personalizzati con codice VBA](#).

Panoramica delle celle da salvare

Per visualizzare tutte le celle configurate per il salvataggio, fare clic su "**Cell Save And Restore**" nella scheda della barra multifunzione di XLS Padlock:



Si apre una finestra che elenca tutte le celle configurate. Da qui è possibile rimuovere singole celle oppure cancellare l'intero elenco.



Ripristinare e salvare con VBA

Quando si utilizza la modalità **Save defined cell values only** (salva solo i valori di cella definiti), XLS Padlock consente di salvare e ripristinare valori personalizzati oltre ai valori di cella predefiniti. Ciò è utile per salvare variabili, impostazioni o dati che non sono memorizzati direttamente in una cella.

Panoramica

Questa funzionalità si basa su due eventi VBA e due funzioni dell'API VBA. Il codice viene inserito nelle subroutine di evento per leggere o scrivere i dati personalizzati quando l'utente carica o salva il proprio lavoro.

Eventi e funzioni principali

Eventi VBA

Le due subroutine di evento seguenti devono essere inserite in un modulo della cartella di lavoro Excel. XLS Padlock le richiamerà automaticamente durante il processo di salvataggio e caricamento.

```
' Called when a user loads a save file.
Sub XLSPadlock_RestoreCustomValues()
    ' Your code to read values goes here.
    MsgBox ("Restoring custom values...")
End Sub

' Called when a user saves their work.
Sub XLSPadlock_SaveCustomValues()
    ' Your code to write values goes here.
    MsgBox ("Saving custom values...")
End Sub
```

Funzioni dell'API VBA

- **WriteCustomCellValue(UniqueID, Value)**: scrive un singolo valore stringa associato a un identificatore univoco.
- **ReadCustomCellValue(UniqueID, DefaultValue)**: legge un singolo valore stringa per un dato identificatore. Se l'identificatore non viene trovato, la funzione restituisce `DefaultValue`.

Importante

`WriteCustomCellValue` e `ReadCustomCellValue` funzionano solo quando vengono richiamate rispettivamente dagli eventi `XLSPadlock_SaveCustomValues` e `XLSPadlock_RestoreCustomValues`.

Letture e scrittura di valori singoli

Ecco come scrivere un singolo valore:

```
Sub XLSPadlock_SaveCustomValues()  
    Dim XLSPadlock1 As Object  
    On Error Resume Next  
    Set XLSPadlock1 = Application.COMAddIns("GXLS.GXLSPLock").Object  
    XLSPadlock1.WriteCustomCellValue "MySetting", "MyValue"  
End Sub
```

Ed ecco come rileggerlo:

```
Sub XLSPadlock_RestoreCustomValues()  
    Dim XLSPadlock1 As Object  
    On Error Resume Next  
    Set XLSPadlock1 = Application.COMAddIns("GXLS.GXLSPLock").Object  
  
    Dim MySettingValue As String  
    MySettingValue = XLSPadlock1.ReadCustomCellValue("MySetting", "Default")  
End Sub
```

Letture di tutti i valori come dizionario

È inoltre possibile leggere tutti i valori personalizzati salvati in una sola volta passando una stringa vuota come identificatore a `ReadCustomCellValue`. Ciò restituisce un oggetto `Scripting.Dictionary`.

Riferimento richiesto

Per utilizzare questa funzionalità, è necessario aggiungere un riferimento a **"Microsoft Scripting Runtime"** nel progetto VBA (dal menu della VBE, andare in Tools -> References).

```
``vb Sub XLSPadlock_RestoreCustomValues() Dim XLSPadlock1 As Object On Error Resume Next Set  
XLSPadlock1 = Application.COMAddIns("GXLS.GXLSPLock").Object
```

```
Dim Dict As Object ' Scripting.Dictionary  
Set Dict = XLSPadlock1.ReadCustomCellValue("", "")  
  
If Not Dict Is Nothing Then  
    For Each Key In Dict.Keys  
        MsgBox "Key: " & Key & ", Value: " & Dict(Key)  
    Next Key  
End If
```

End Sub

```
## Esempio completo: salvataggio/ripristino di una colonna
```

```
<div class="aside aside-note">
```

```
L'intero codice seguente deve essere inserito in un unico modulo.
```

```
</div>
```

Questa funzione di supporto genera una stringa separata da virgole a partire dai valori di un c

```
```\vb
Function CsvRange(myRange As Range) As String
 Dim csvRangeOutput As String
 Dim entry As Variant
 For Each entry In myRange
 If Not IsEmpty(entry.Value) Then
 csvRangeOutput = csvRangeOutput & entry.Value & ","
 End If
 Next
 If Len(csvRangeOutput) > 0 Then
 CsvRange = Left(csvRangeOutput, Len(csvRangeOutput) - 1)
 End If
End Function
```

Questo evento viene richiamato quando l'utente salva il proprio lavoro. Utilizza la funzione di supporto per convertire l'intero intervallo utilizzato della colonna A in un'unica stringa e la salva.

```
Sub XLSPadlock_SaveCustomValues()
 Dim XLSPadlock1 As Object
 On Error Resume Next
 Set XLSPadlock1 = Application.COMAddIns("GXLS.GXLSPLock").Object

 Dim rng As Range
 Set rng = ThisWorkbook.Worksheets(2).Range("A1").CurrentRegion

 Dim myString As String
 myString = CsvRange(rng)

 XLSPadlock1.WriteCustomCellValue "MyEntireColumnA", myString
End Sub
```

Questo evento viene richiamato quando l'utente carica un file di salvataggio. Legge la stringa e ripristina i valori nella colonna A.

```
Sub XLSPadlock_RestoreCustomValues()
 Dim XLSPadlock1 As Object
 On Error Resume Next
 Set XLSPadlock1 = Application.COMAddIns("GXLS.GXLSPLock").Object

 Dim Val As String
 Val = XLSPadlock1.ReadCustomCellValue("MyEntireColumnA", "")

 If Val <> "" Then
 Dim r As Range, i As Long, ar
 Set r = ThisWorkbook.Worksheets(2).Range("A:A")
 r.ClearContents

 ar = Split(Val, ",")
 For i = 0 To UBound(ar)
 r.Cells(i + 1, 1).Value = ar(i)
 Next
 End If
End Sub

See Also

- [How to migrate user data from a previous version](#chapter-excel-vba-migrate-user-data-updat
```

# Accedere alla cartella di lavoro protetta e ai file complementari

---

## NOTA

Questa guida spiega come avviare una nuova istanza di Excel dalla cartella di lavoro protetta e concederle l'accesso alla cartella di lavoro sicura e ai relativi file companion.

Per impostazione predefinita, solo l'istanza di Excel avviata dall'EXE compilato può accedere alla cartella di lavoro sicura e ai relativi file associati. Se si crea una nuova istanza di Excel utilizzando VBA, questa non avrà il permesso di aprire tali file protetti.

Per concedere l'accesso, occorre utilizzare l'API di XLS Padlock allo scopo di trasmettere l'handle della finestra (HWND) della nuova istanza di Excel all'applicazione principale. In questo modo si autorizza la nuova istanza a lavorare con il file system virtuale.

```

Sub RunInSeparateExcelInstance()
 Dim XLSPadlock As Object
 Dim appExcel As Object ' Excel.Application
 Dim wbExcel As Object ' Excel.Workbook
 Dim companionFilePath As String

 ' Use LongPtr for 64-bit compatibility when getting the window handle.
 #If VBA7 Then
 Dim windowHandle As LongPtr
 #Else
 Dim windowHandle As Long
 #End If

 On Error GoTo Cleanup

 ' Get the path to a companion file located in the same folder as the EXE.
 companionFilePath = GetPathToFileInEXEFolder("data.xlsx")
 If companionFilePath = "" Then
 MsgBox "Companion file not found.", vbExclamation
 Exit Sub
 End If

 ' Get the XLS Padlock API object.
 Set XLSPadlock = Application.COMAddIns("GXLS.GXLSPLock").Object
 If XLSPadlock Is Nothing Then
 MsgBox "XLS Padlock API not available. Is the workbook protected?", vbCritical
 Exit Sub
 End If

 ' Start a new, invisible Excel instance.
 Set appExcel = CreateObject("Excel.Application")

 ' Get the window handle (HWND) of the new Excel instance.
 windowHandle = appExcel.Application.Hwnd

 ' Authorize the new instance by passing its HWND to the main application.
 ' Option "5" is used for this purpose.
 XLSPadlock.SetOption Option:="5", Value:=windowHandle

 ' The new Excel instance can now access the secure companion file.
 Set wbExcel = appExcel.Workbooks.Open(companionFilePath, False, True)

 ' Example: Read a value from the companion workbook.
 MsgBox "Value from companion file: " & wbExcel.Worksheets("Sheet1").Cells(1, 1).Value

Cleanup:
 If Err.Number <> 0 Then
 MsgBox "An error occurred: " & Err.Description, vbCritical
 End If

 ' Close the workbook and quit the new Excel instance.
 If Not wbExcel Is Nothing Then wbExcel.Close SaveChanges:=False

```

```
If Not appExcel Is Nothing Then appExcel.Quit

' Release object variables.
Set wbExcel = Nothing
Set appExcel = Nothing
Set XLSPadlock = Nothing
End Sub

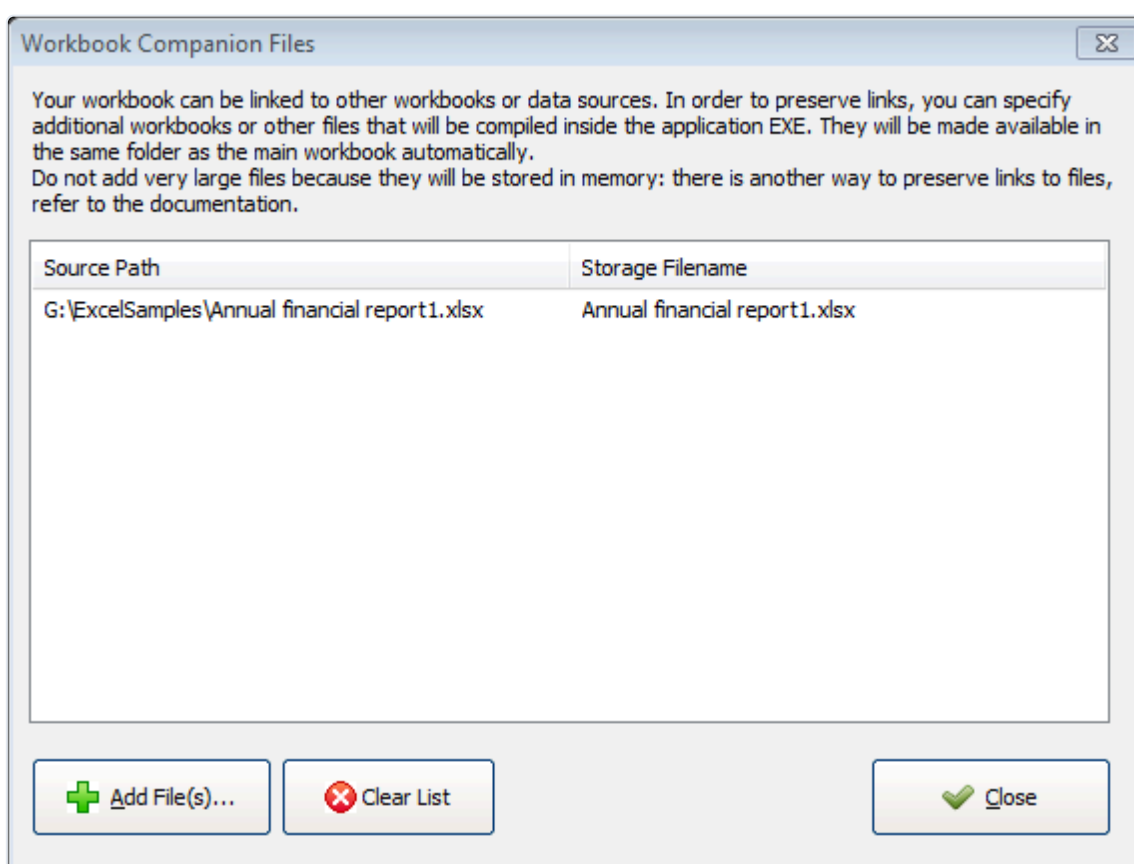
Public Function GetPathToFileInEXEFolder(ByVal Filename As String) As String
' Helper function to get the full path to a file in the EXE's directory.
' See: https://www.xlspadlock.com/doc/get-the-path-to-a-file-in-the-same-folder-as-the-comp
Dim XLSPadlock As Object
On Error Resume Next
Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
If Not XLSPadlock Is Nothing Then
 GetPathToFileInEXEFolder = Application.BuildPath(XLSPadlock.PLEvalVar("EXEPath"), Filename)
End If
End Function
```

# Aggiungere file complementari

Se la cartella di lavoro è collegata ad altre cartelle di lavoro o origini dati, è possibile aggiungerle come "file companion". Questo le raggruppa direttamente nell'EXE dell'applicazione principale, semplificando la distribuzione. Quando l'applicazione viene eseguita, questi file vengono resi disponibili nella stessa cartella virtuale della cartella di lavoro principale.

È possibile includere anche i componenti aggiuntivi XLL.

Ad esempio, se la cartella di lavoro utilizza un file di testo esterno come origine dati, aggiungerlo come file companion garantisce che venga sempre trovato quando la cartella di lavoro compilata viene aperta su qualsiasi computer.



Per aggiungere file, fare clic su **Add Files** (aggiungi file) e selezionarli. Verranno visualizzati nell'elenco. La colonna "Storage Filename" (nome file di archiviazione) mostra il nome del file che verrà utilizzato quando il file viene compilato nell'EXE e ripristinato in fase di esecuzione.

## Evitare i file di grandi dimensioni

Non aggiungere file molto grandi, poiché verranno memorizzati interamente in memoria. Per collegarsi ad altre cartelle di lavoro o file di dati di grandi dimensioni, è preferibile collocarli nella stessa cartella dell'EXE dell'applicazione e farvi riferimento utilizzando la funzione `PLEvalVar("XLSPath")`. Consultare [Utilizzare riferimenti esterni e collegamenti ipertestuali](#) per maggiori dettagli.

### ### Accedere ai file companion con VBA

Per accedere ai file companion con VBA, è possibile utilizzare la seguente funzione:

```
Public Function PathToCompiledFile(Filename As String) As String
 Dim XLSPadlock As Object
 On Error GoTo Err

 Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
 PathToCompiledFile = XLSPadlock.PLEvalVar("XLSPath") & Filename

 Exit Function
Err:
 PathToCompiledFile = ""
End Function
```

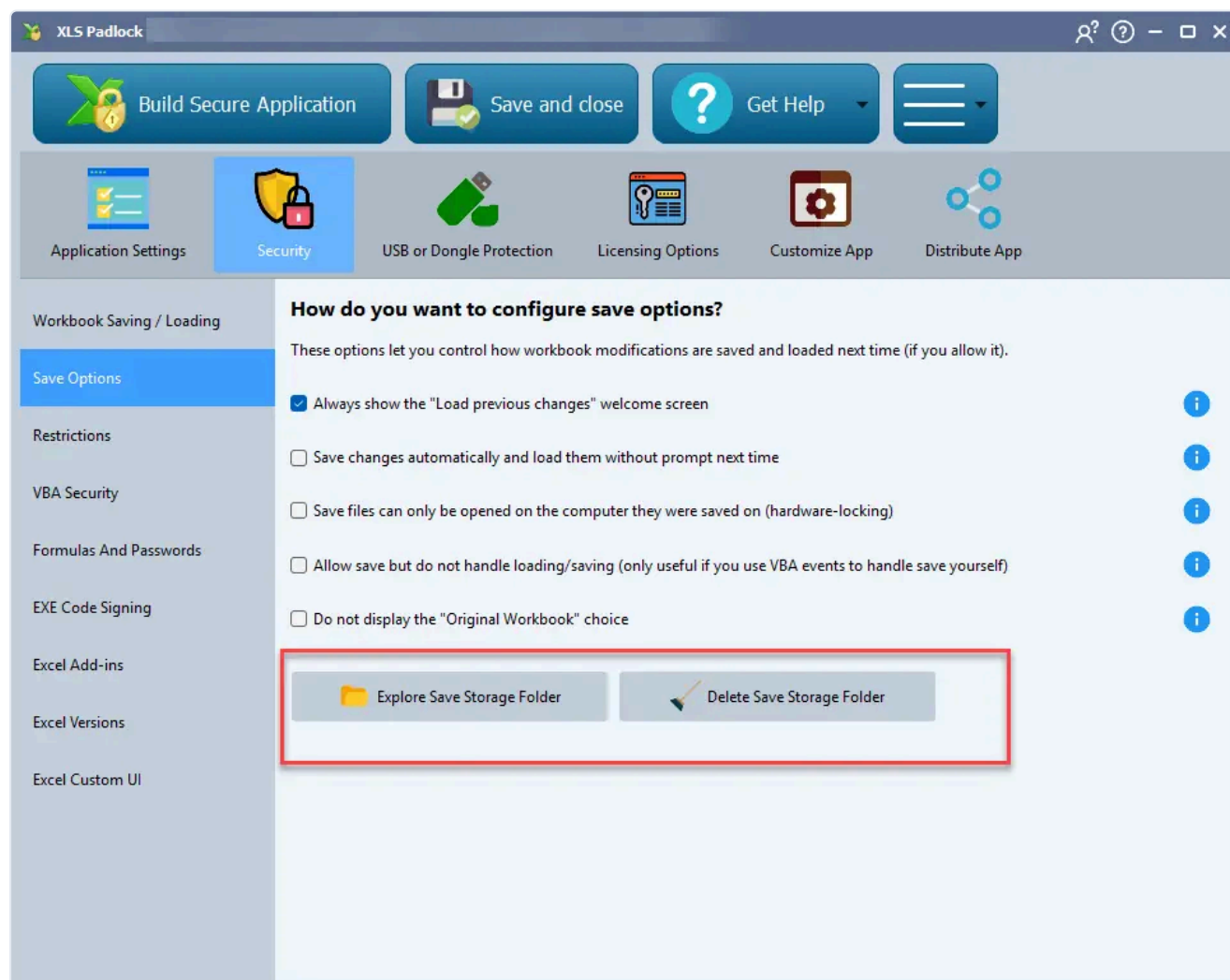
È quindi possibile utilizzare questa funzione per aprire un file companion:

```
Sub ExampleUsage()
 Dim wk As Workbook
 Dim filePath As String

 filePath = PathToCompiledFile("Test File.xlsx")
 If filePath <> "" Then
 Set wk = Workbooks.Open(filePath, False, False)
 MsgBox wk.Sheets(1).Cells(1, 1).Value
 wk.Close
 Else
 MsgBox "Companion file not found!"
 End If
End Sub
```

# Cartella di archiviazione dei salvataggi

I pulsanti **Explore Save Storage Folder** e **Delete Save Storage Folder** offrono un accesso rapido alla cartella locale del PC in cui l'applicazione protetta archivia i propri dati.



Questa cartella contiene: i file salvati automaticamente (se questa funzionalità è abilitata), il file della cronologia dei salvataggi, i dati di licenza e attivazione.

Se la modalità portatile è abilitata (nelle [Opzioni avanzate](#)), questi pulsanti apriranno la stessa cartella che contiene il file EXE dell'applicazione. Per sicurezza, il pulsante "Delete" è disabilitato in questa modalità.

# Archiviare le modifiche all'interno dell'EXE

---

No, gli utenti non possono salvare le modifiche direttamente nel file `.exe` compilato. Modificare un file eseguibile dopo la compilazione può corromperlo e spesso provoca falsi positivi da parte dei software antivirus.

Per questo motivo, XLS Padlock è progettato per salvare i dati utente in file di salvataggio cifrati separati (con estensioni `.xlsc` o `.xlsce`), e non nell'EXE principale dell'applicazione.

Si consideri l'EXE compilato come l'applicazione stessa (come `Excel.exe`), e i file di salvataggio come i documenti su cui essa lavora.

# Limitare il caricamento e il salvataggio

Una potenziale vulnerabilità è che le cartelle di lavoro Excel possono essere salvate su disco tramite comandi VBA o OLE. Per impedire questo metodo di estrazione dei dati, XLS Padlock include una potente opzione di sicurezza denominata **"Do not allow loading/saving other workbooks"** (Vietare il caricamento/salvataggio di altre cartelle di lavoro).

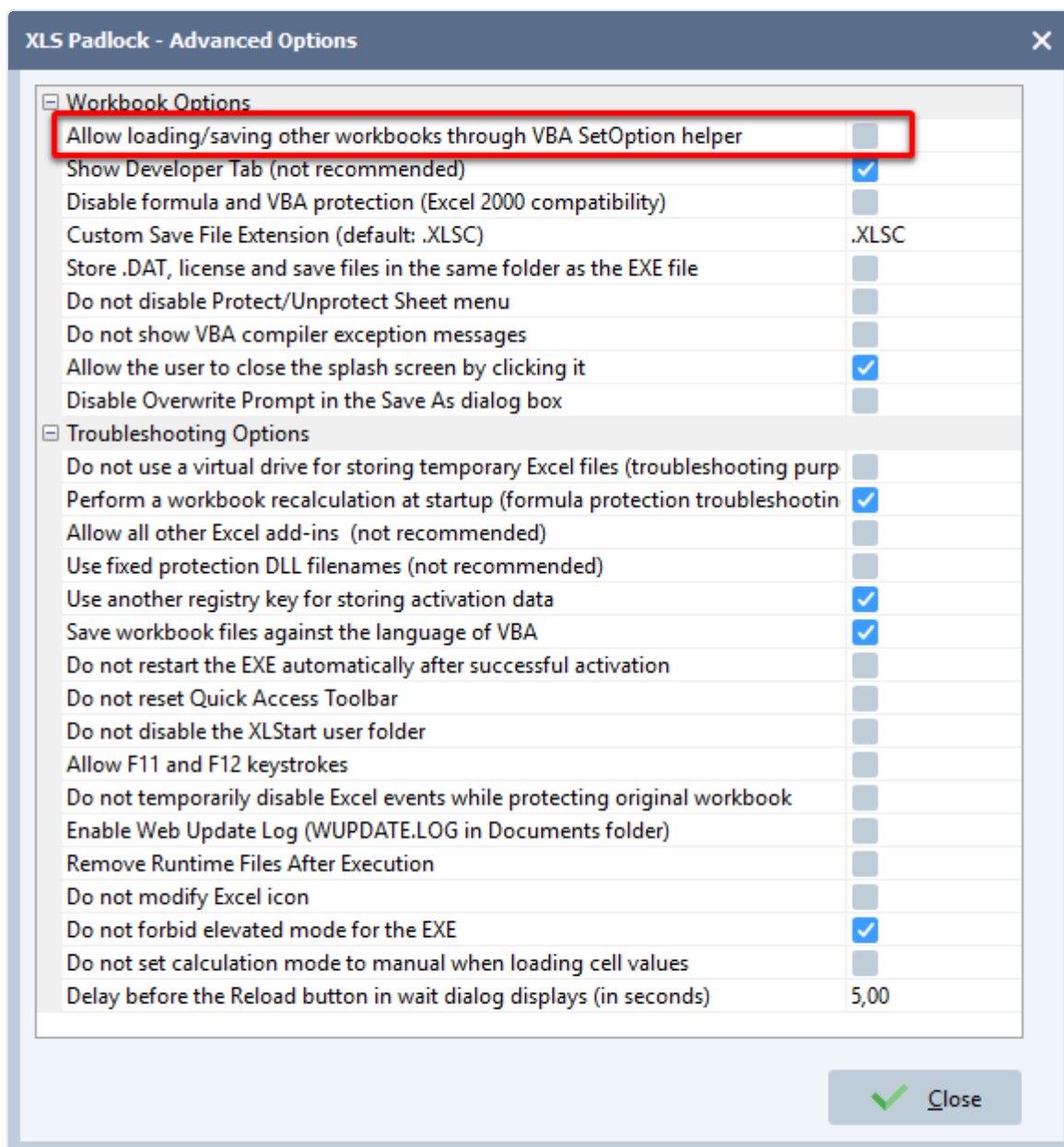
Quando è attivata, questa funzionalità rende l'istanza di Excel che esegue l'applicazione **incapace di caricare o salvare qualsiasi altro file di cartella di lavoro**. Ciò impedisce efficacemente agli utenti di aprire altre cartelle di lavoro e blocca gli hack basati su VBA che tentano di copiare i dati in un nuovo file non protetto.



## Cosa fare se la mia applicazione deve salvare altre cartelle di lavoro? La soluzione VBA SetHelper

Questa funzionalità di sicurezza blocca anche il codice VBA standard che salva o carica cartelle di lavoro. Se l'applicazione deve legittimamente eseguire queste azioni, è disponibile una soluzione alternativa.

È possibile disattivare temporaneamente la restrizione all'interno del proprio codice VBA. Per farlo, occorre innanzitutto attivare l'opzione avanzata **"Allow loading/saving other workbooks through VBA SetOption helper"** (Consentire il caricamento/salvataggio di altre cartelle di lavoro tramite l'helper VBA SetOption) qui:



Successivamente, è possibile utilizzare il seguente codice VBA:

```
Dim XLSPadlock As Object
Set XLSPadlock = Application.COMAddIns("GXLS.GXLSPLock").Object

' Temporarily disable the security
XLSPadlock.SetOption Option:= "2", Value:= "0"
XLSPadlock.SetOption Option:= "1", Value:= "1" ' Also disable encrypted save prompt

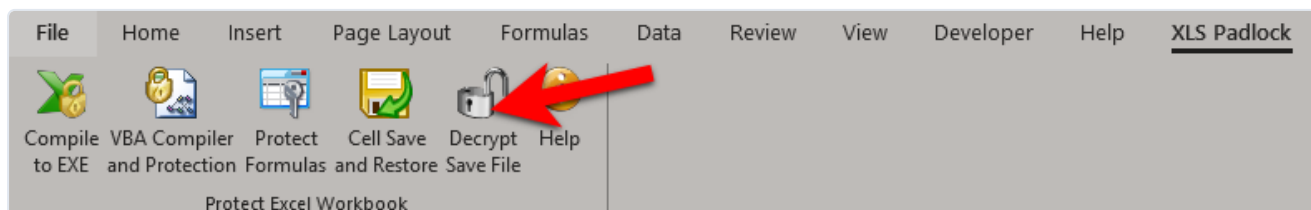
' Your code to save a normal workbook
ActiveWorkbook.SaveAs "D:\My Documents\NormalWorkbook.xlsx"

' Re-enable the security
XLSPadlock.SetOption Option:= "2", Value:= "1"
XLSPadlock.SetOption Option:= "1", Value:= "0"
```

# Aprire e decifrare i salvataggi

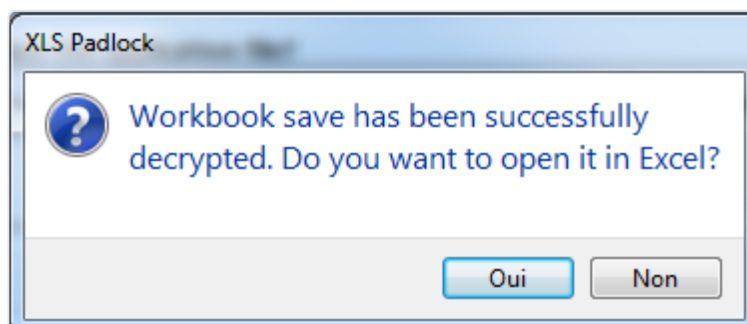
I file di salvataggio creati dalla vostra applicazione sono cifrati e non possono essere aperti direttamente in Excel. Tuttavia, in qualità di autore originale della cartella di lavoro, potete decifrare qualsiasi file di salvataggio generato dalla vostra applicazione. Questo è utile per recuperare i dati se un cliente vi invia il proprio file di salvataggio.

Per aprire un file di salvataggio ( .xlsc o .xlsce ), scegliete "Decrypt Save File" (Decifra file di salvataggio) dalla barra multifunzione di XLS Padlock in Excel:



## Decifratura dei file .XLSC (modalità di salvataggio completo)

Se selezionate un file .xlsc creato con la modalità di salvataggio completo, XLS Padlock decifrerà l'intera cartella di lavoro e la aprirà.



### Limiti della decifratura

- Le formule protette con XLS Padlock non possono essere recuperate durante la decifratura di un file di salvataggio. La cartella di lavoro decifrata potrebbe non essere completamente funzionale. Questa funzionalità dovrebbe essere utilizzata solo per recuperare i dati inseriti dall'utente.
- Lo stesso limite si applica al progetto VBA se lo avete bloccato o compilato.
- Per utilizzare la cartella di lavoro con tutte le funzionalità intatte, dovrete eseguire la vostra applicazione protetta e caricare il file di salvataggio normalmente.

## Decifratura dei file .XLSCE (modalità valori delle celle)

Se selezionate un file .xlsce creato con la modalità di salvataggio dei valori delle celle, XLS Padlock vi chiederà se desiderate sovrascrivere le celle della cartella di lavoro attualmente aperta con i valori del file di salvataggio.

**Questa operazione sovrascriverà i vostri dati**

La conferma di questa operazione sovrascriverà definitivamente i valori delle celle nella cartella di lavoro sorgente aperta.

**File di progetto richiesto**

Questa funzionalità di decifratura richiede il file di progetto XLS Padlock originale ( .xplp ) utilizzato per creare l'applicazione. XLS Padlock utilizza le impostazioni del vostro progetto (come la Secret Key) per cifrare e decifrare i file di salvataggio. Senza il file di progetto corretto, non potete decifrare i salvataggi. **Non condividete mai i vostri file di progetto XLS Padlock.**

# Bloccare i salvataggi su una macchina

---

Questa opzione impedisce agli utenti finali di condividere i loro file di salvataggio. Quando è attivata, l'applicazione crea **file di salvataggio bloccati a livello hardware** (hardware-locked save files), che possono essere aperti solo sul computer su cui sono stati creati. Se un utente tenta di aprire un file di salvataggio su un computer diverso, il caricamento non riuscirà.

L'applicazione incorpora nel file di salvataggio l'**ID di sistema** univoco del computer e lo verifica al momento del caricamento.

## SUGGERIMENTO

In qualità di proprietario dell'applicazione, è comunque possibile decifrare autonomamente qualsiasi file di salvataggio bloccato a livello hardware utilizzando la funzione **Decrypt Save File** (decifra file di salvataggio) di XLS Padlock.

## Avvertenza

Questa funzionalità è disponibile solo se è stata selezionata la modalità Full Save (salvataggio completo).

# Riferimenti esterni e collegamenti ipertestuali

XLS Padlock protegge una cartella di lavoro per ogni file EXE. Se la cartella di lavoro utilizza riferimenti esterni o richiede file aggiuntivi, è necessario aggiornare i percorsi affinché funzionino correttamente dopo la compilazione.

XLS Padlock offre due modi principali per gestire i file esterni:

- Aggiungerli come [Companion Files](#).
- Utilizzare percorsi dinamici nei collegamenti ipertestuali o nel codice VBA.

Per gestire i percorsi dinamici, XLS Padlock offre una funzione chiamata `PLEvalVar` che può essere utilizzata direttamente nelle formule di Excel oppure richiamata da VBA.

Questa funzione accetta un argomento di tipo stringa:

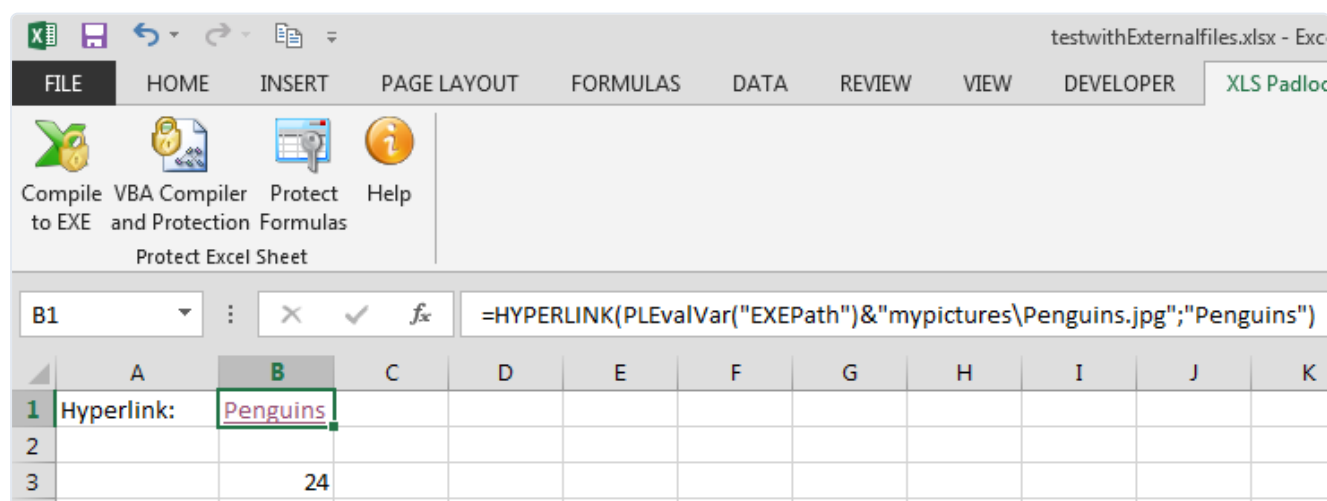
- `=PLEvalVar("EXEPath")` restituisce il percorso completo della cartella che contiene il file EXE dell'applicazione (con la barra rovesciata finale).
- `=PLEvalVar("XLSPath")` restituisce il percorso completo della cartella che contiene la cartella di lavoro compilata in fase di esecuzione (con la barra rovesciata finale).

## NOTA

Questa cartella è una cartella virtuale e, di conseguenza, non è possibile collocarvi file reali. È utile solo se si lavora con i Companion files (vedere [Add Companion Files](#)).

## Esempio 1

Si dispone di collegamenti ipertestuali verso file immagine esterni. Questi file immagine si trovano nella stessa cartella del file XLS della cartella di lavoro (o in una sottocartella).



Si dispone di un collegamento ipertestuale in una cella, definito da:

```
=HYPERLINK("Penguins.jpg", "Penguins")
```

Per farlo funzionare con XLS Padlock, è necessario copiare tutti i file immagine esterni nella stessa cartella del file EXE generato con XLS Padlock. Successivamente, occorre modificare tutti i collegamenti ipertestuali per inserirvi la funzione `PLEvalVar("EXEPath")`, che restituisce il percorso di tale cartella.

Nel nostro caso, diventerà:

```
=HYPERLINK(PLEvalVar("EXEPath") & "Penguins.jpg", "Penguins")
```

### Avvertenza

I file esterni devono essere distribuiti nella stessa cartella del file EXE finale dell'applicazione. È inoltre buona prassi evitare gli spazi nei nomi dei file.

Ciò funziona anche per i file contenuti nelle sottocartelle. Un collegamento come:

```
=HYPERLINK("My Pictures\\Penguins.jpg", "Penguins")
```

...deve essere modificato in:

```
=HYPERLINK(PLEvalVar("EXEPath") & "mypictures\\Penguins.jpg", "Penguins")
```

## Esempio 2

Per accedere ai file esterni con VBA, è possibile utilizzare una funzione di supporto che costruisce il percorso completo verso un file situato nella stessa cartella dell'EXE.

👉 Per una spiegazione dettagliata e uno snippet di codice riutilizzabile, consultare la guida su come ottenere il percorso di un file situato nella cartella dell'EXE.

```
Public Function GetPathToFileInEXEFolder(ByVal Filename As String) As String
 Dim XLSPadlock As Object
 Dim exePath As String
 On Error GoTo Err

 Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
 exePath = XLSPadlock.PLEvalVar("EXEPath")
 GetPathToFileInEXEFolder = Application.BuildPath(exePath, Filename)
 Exit Function
Err:
 GetPathToFileInEXEFolder = ""
End Function
```

# Ottenere il percorso accanto alla cartella di lavoro compilata

---

Questa funzione VBA recupera il percorso completo di un file che si trova nella stessa directory dell'EXE della propria applicazione. È particolarmente utile per accedere a risorse esterne o a file companion distribuiti insieme alla cartella di lavoro protetta.

👉 Inserire la seguente funzione in un modulo VBA:

```
Public Function PathToFile(ByVal Filename As String) As String
 Dim XLSPadlock As Object
 Dim exePath As String
 On Error GoTo Err

 Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object
 exePath = XLSPadlock.PLEvalVar("EXEPath")

 ' Use Application.BuildPath to correctly join the path and filename
 PathToFile = Application.BuildPath(exePath, Filename)

 Exit Function
Err:
 PathToFile = ""
End Function
```

È quindi possibile richiamare la funzione:

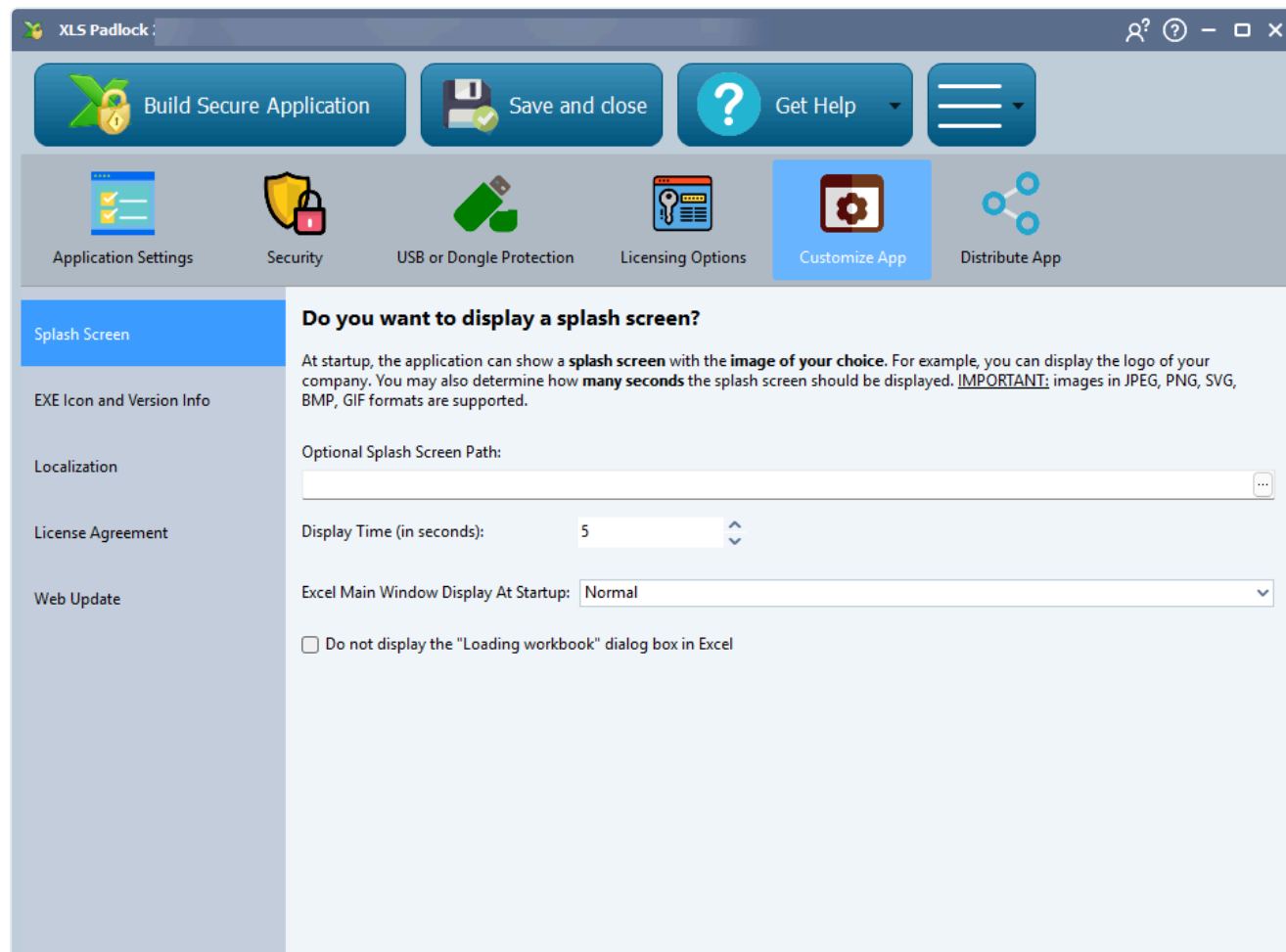
```
Sub Test_File()
 DoSomethingWith(PathToFile("data.xls"))
End Sub
```

👉 Vedere anche: [VBA API Cookbook & Recipes](#)

# Personalizzare l'applicazione

XLS Padlock offre diversi modi per **personalizzare l'applicazione Excel compilata**, consentendo di adattarne l'aspetto, i messaggi e il comportamento per gli utenti finali.

Questo argomento presenta una panoramica delle opzioni disponibili in "**Customize App**" in XLS Padlock:



## Schermata iniziale

Mostrare una **schermata iniziale** personalizzata all'avvio dell'applicazione, ad esempio il logo della propria azienda o un'immagine di benvenuto a marchio personalizzato.

È possibile scegliere il formato dell'immagine (JPEG, PNG, BMP, SVG o GIF), definire per quanti secondi rimane visibile e controllare il comportamento della finestra di Excel durante l'avvio.

[➔ Maggiori informazioni sulle impostazioni della schermata iniziale](#)

## Icona dell'EXE e informazioni sulla versione

Sostituire l'icona predefinita di XLS Padlock con la **propria icona dell'applicazione**.

È inoltre possibile specificare informazioni sulla versione come il **nome del prodotto**, il **nome dell'azienda**, il **copyright** e la **descrizione del file**, tutte visibili nella finestra di dialogo delle proprietà del

file di Windows.

Ciò consente di dare al proprio EXE l'immagine di un'applicazione autonoma professionale.

→ [Maggiori informazioni sull'icona dell'EXE](#) e sulle [informazioni sulla versione](#)

## Localizzazione

Personalizzare o tradurre tutte le **finestre di dialogo, i messaggi e le richieste integrati** visualizzati dalla cartella di lavoro protetta.

XLS Padlock fornisce un sistema di localizzazione flessibile che consente di adattare l'applicazione a lingue diverse o di creare messaggi utente completamente personalizzati.

→ [Maggiori informazioni sulla localizzazione](#)

## Contratto di licenza

Aggiungere un **contratto di licenza** che gli utenti devono accettare prima di eseguire l'applicazione Excel compilata per la prima volta.

È possibile includere testo semplice o caricare un file RTF/HTML esterno.

Se gli utenti rifiutano il contratto, l'applicazione non proseguirà con l'apertura.

→ [Maggiori informazioni sul contratto di licenza](#)

## Aggiornamento Web

Attivare gli **aggiornamenti basati sul Web** per l'applicazione Excel.

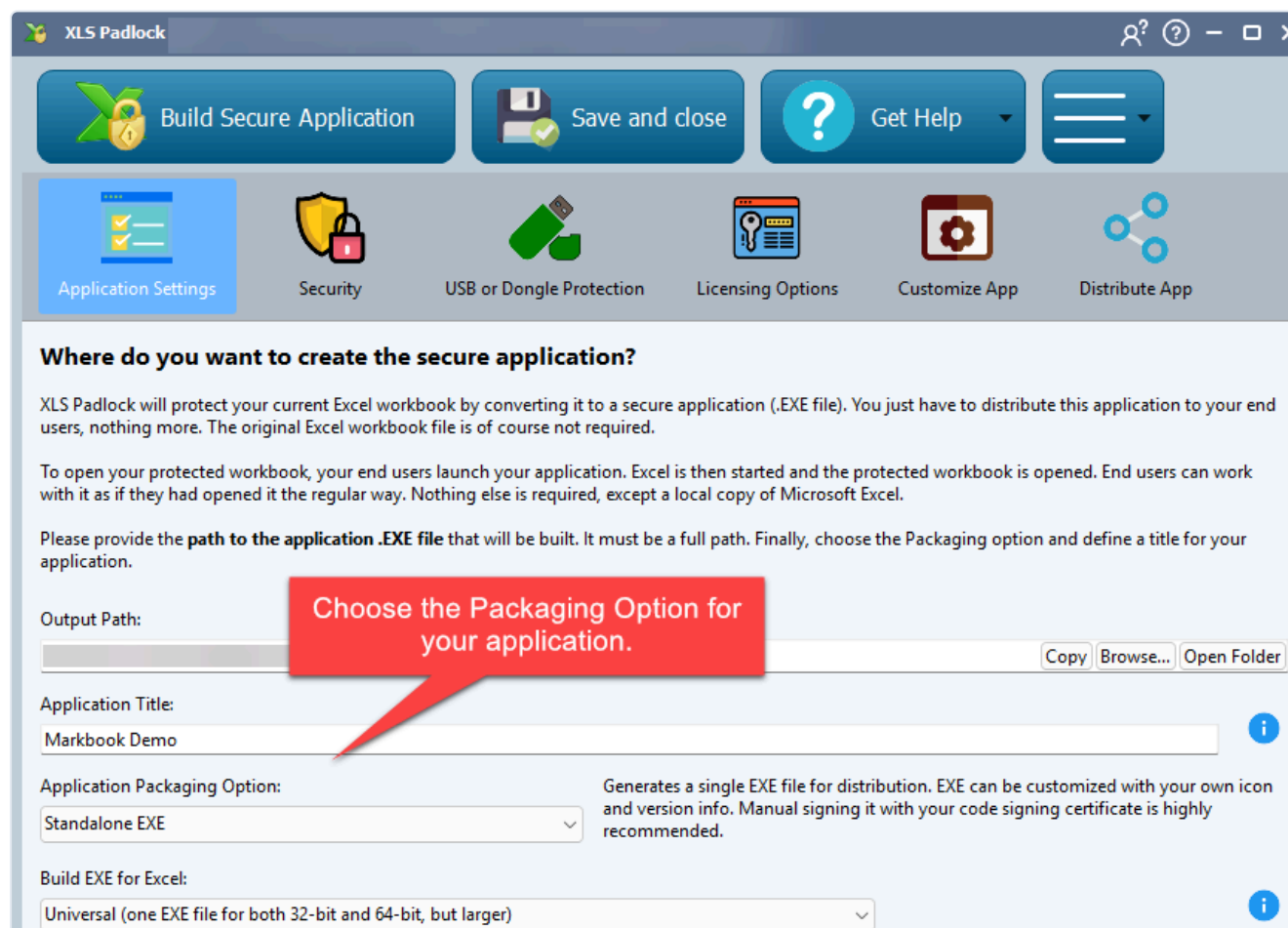
Questa funzionalità consente di avvisare gli utenti quando è disponibile una nuova versione del proprio EXE e di scaricarla automaticamente dal proprio sito Web o server.

È un modo efficiente per mantenere aggiornati gli utenti con il minimo sforzo.

→ [Maggiori informazioni sull'aggiornamento Web](#)

# Pacchettizzazione dell'applicazione

L'opzione **Application Packaging** determina la modalità di compilazione della cartella di lavoro protetta, offrendo due formati distinti.



**Where do you want to create the secure application?**

XLS Padlock will protect your current Excel workbook by converting it to a secure application (.EXE file). You just have to distribute this application to your end users, nothing more. The original Excel workbook file is of course not required.

To open your protected workbook, your end users launch your application. Excel is then started and the protected workbook is opened. End users can work with it as if they had opened it the regular way. Nothing else is required, except a local copy of Microsoft Excel.

Please provide the **path to the application .EXE file** that will be built. It must be a full path. Finally, choose the Packaging option and define a title for your application.

Output Path:  Copy Browse... Open Folder

Application Title:  i

Application Packaging Option:  v Generates a single EXE file for distribution. EXE can be customized with your own icon and version info. Manual signing it with your code signing certificate is highly recommended.

Build EXE for Excel:  v i

## Raccomandazione

Si consiglia l'opzione **Standalone EXE** se si dispone di un [certificato di firma del codice](#). Sebbene non sia obbligatoria, la firma dell'EXE è il modo migliore per evitare gli avvisi "Unknown App" di Windows SmartScreen e per ridurre i falsi positivi dei software antivirus.

## Standalone EXE

Questa opzione compila la cartella di lavoro Excel in un unico file eseguibile ( .exe ) autonomo. È possibile distribuire questo singolo file agli utenti, che non avranno bisogno della cartella di lavoro Excel originale.

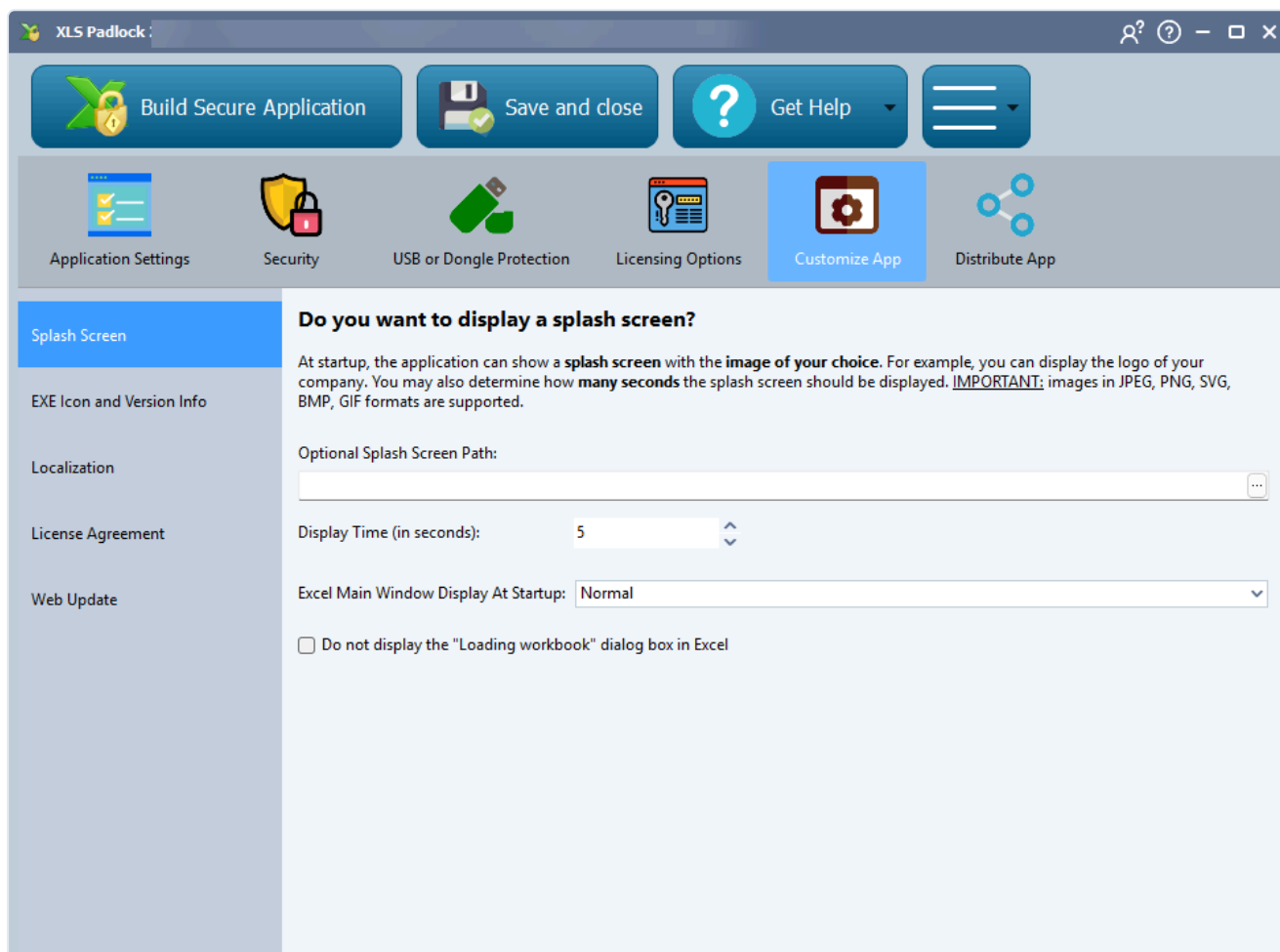
## EXE + XPLAPP Application Bundle

Questa opzione crea un file EXE insieme a un file complementare `.bin64` e a un file di dati `.xplapp`. Il file EXE principale è firmato in anticipo dalla nostra società, il che lo aiuta a essere considerato più affidabile da Windows SmartScreen e dalle soluzioni antivirus.

- Questo metodo di distribuzione richiede che i file EXE, `.bin64` e `.xplapp` siano tutti conservati nella stessa cartella.
- Per comodità, XLS Padlock può raggruppare questi file in un [unico archivio Zip](#) o in un [programma di installazione](#).
- L'utilizzo di questo metodo riduce notevolmente le probabilità di incontrare avvisi "Unknown Application" e falsi positivi degli antivirus.
- Il file complementare `.bin64` garantisce la compatibilità con le versioni a 64 bit di Excel.

# Schermata iniziale

È possibile visualizzare uno **splash screen**, un'immagine mostrata brevemente all'avvio mentre l'applicazione si inizializza. Si tratta di un ottimo modo per mostrare il logo della propria azienda o per personalizzare l'applicazione con il proprio marchio.



## Impostazioni dello splash screen

È possibile utilizzare qualsiasi immagine in formato JPEG, PNG, BMP, SVG o GIF.

È possibile impostare la durata di visualizzazione dello splash screen. Per impostazione predefinita, gli utenti possono chiuderlo immediatamente facendovi clic sopra. Questo comportamento può essere disattivato nelle [Opzioni avanzate](#).

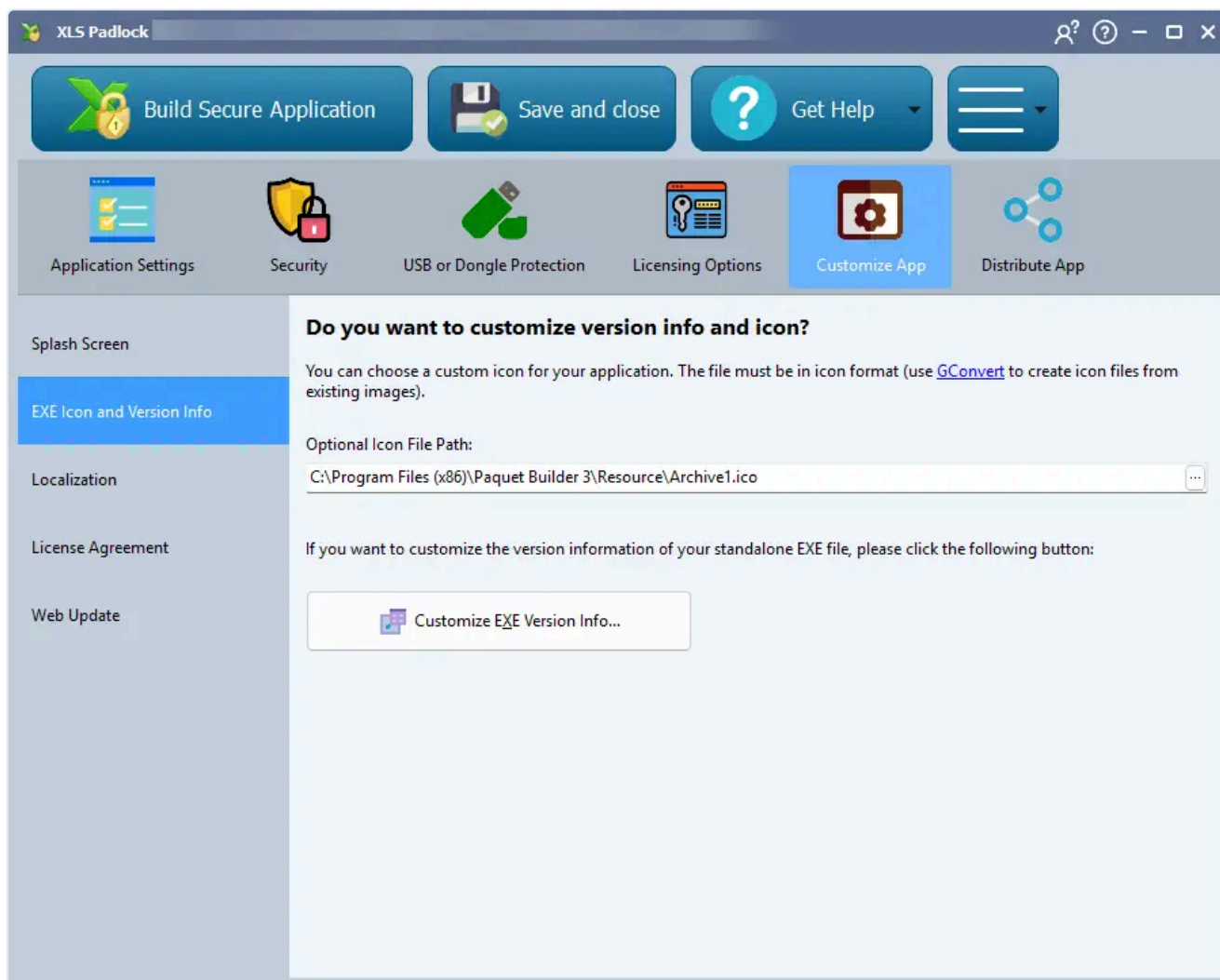
### Utilizzare PNG trasparenti per un aspetto unico

XLS Padlock supporta splash screen non rettangolari e semitrasparenti. Utilizzare un file PNG a 32 bit con un canale alfa per conferire all'applicazione un aspetto moderno e personalizzato.

👉 Vedere anche: [Non visualizzare la finestra di dialogo "Loading workbook" in Excel](#chapter-do-not-display-the-loading-workbook-dialog-box).

# Cambiare l'icona dell'EXE

XLS Padlock consente di sostituire l'**icona predefinita dell'applicazione** con la propria. Per personalizzare l'aspetto della propria applicazione, indicare il **percorso di un file di icona personalizzato**, che deve avere un'estensione `.ico`:



XLS Padlock supporta i formati di icona standard, incluse varie dimensioni (ad esempio 32x32, 48x48) e profondità di colore. Se si necessita di uno strumento per creare o estrarre icone, consigliamo la nostra utilità complementare, [GConvert](#), che può convertire facilmente immagini in icone.

## Avvertenza

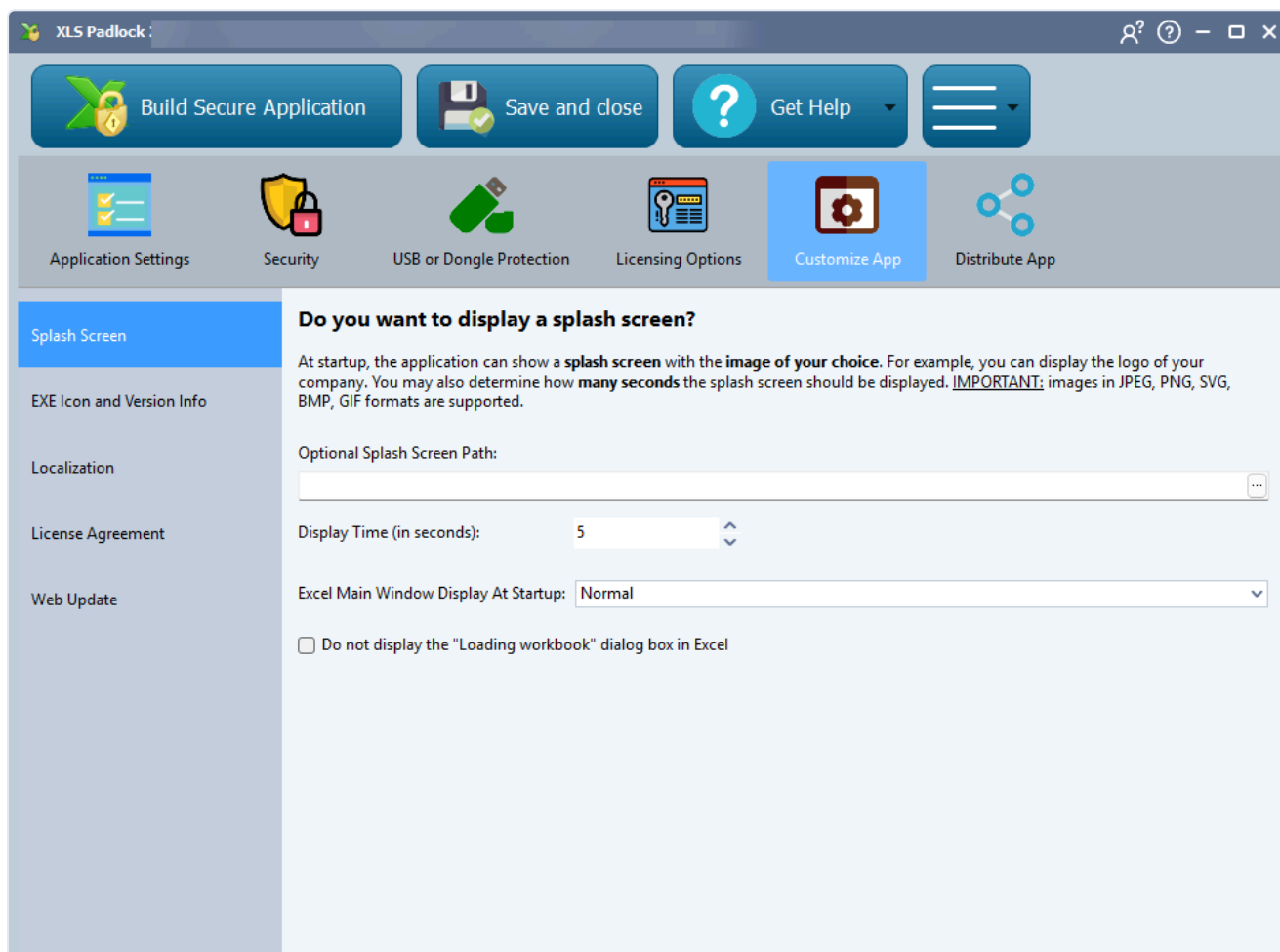
Il file di icona deve essere disponibile come file esterno sul computer durante il processo di compilazione.

**Info**

XLS Padlock applicherà questa icona anche ai file EXE dell'applicazione autonoma, garantendo un'identità di marca coerente.

# Finestra di Excel all'avvio

L'opzione **Excel Main Window Display at Startup** (visualizzazione della finestra principale di Excel all'avvio) consente di definire le dimensioni della finestra di Excel all'avvio dell'applicazione. Questa impostazione si trova nella pagina [Splash Screen](#).



È possibile scegliere fra tre opzioni:

- **Normal:** la finestra di Excel si apre con dimensioni e posizione predefinite. È il comportamento predefinito.
- **Minimized:** l'applicazione si avvia ridotta a icona nella barra delle applicazioni di Windows. È utile se l'applicazione viene eseguita in background o se si desidera visualizzare un modulo personalizzato prima di mostrare la finestra principale di Excel.
- **Maximized:** la finestra di Excel si apre ingrandita per riempire l'intero schermo.

### Occorre nascondere completamente la finestra di Excel?

Se l'applicazione è gestita unicamente da codice VBA e UserForm e si desidera che la finestra di Excel, insieme alla sua icona nella barra delle applicazioni, non compaia mai del tutto, consultare

[Eseguire come applicazione solo VBA \(finestra principale di Excel completamente nascosta\)](#).

Quando questa opzione è attivata, la scelta sopra indicata viene ignorata ed Excel si avvia sempre nascosto.

## Vedere anche

---

- [Come configurare lo Splash Screen](#)
- [Come nascondere la finestra di dialogo "Loading workbook"](#)
- [Eseguire come applicazione solo VBA \(finestra principale di Excel completamente nascosta\)](#)

# Eseguire come applicazione solo VBA

---

Se la cartella di lavoro protetta è guidata interamente da **codice VBA** e **UserForm**, e il foglio di calcolo stesso non è mai destinato a essere visto dagli utenti finali, è possibile distribuirla come *app solo VBA*. La finestra principale di Excel rimane nascosta dall'inizio alla fine, inclusa la sua icona nella barra delle applicazioni, e sullo schermo appare soltanto il proprio UserForm. L'applicazione ha l'aspetto e il comportamento di un programma Windows autonomo, senza alcuna traccia visibile di Excel.

Questa opzione è disponibile a partire da **XLS Padlock 2026.0**.

## Quando utilizzarla

---

Utilizzare la modalità solo VBA quando **tutte** le condizioni seguenti sono soddisfatte:

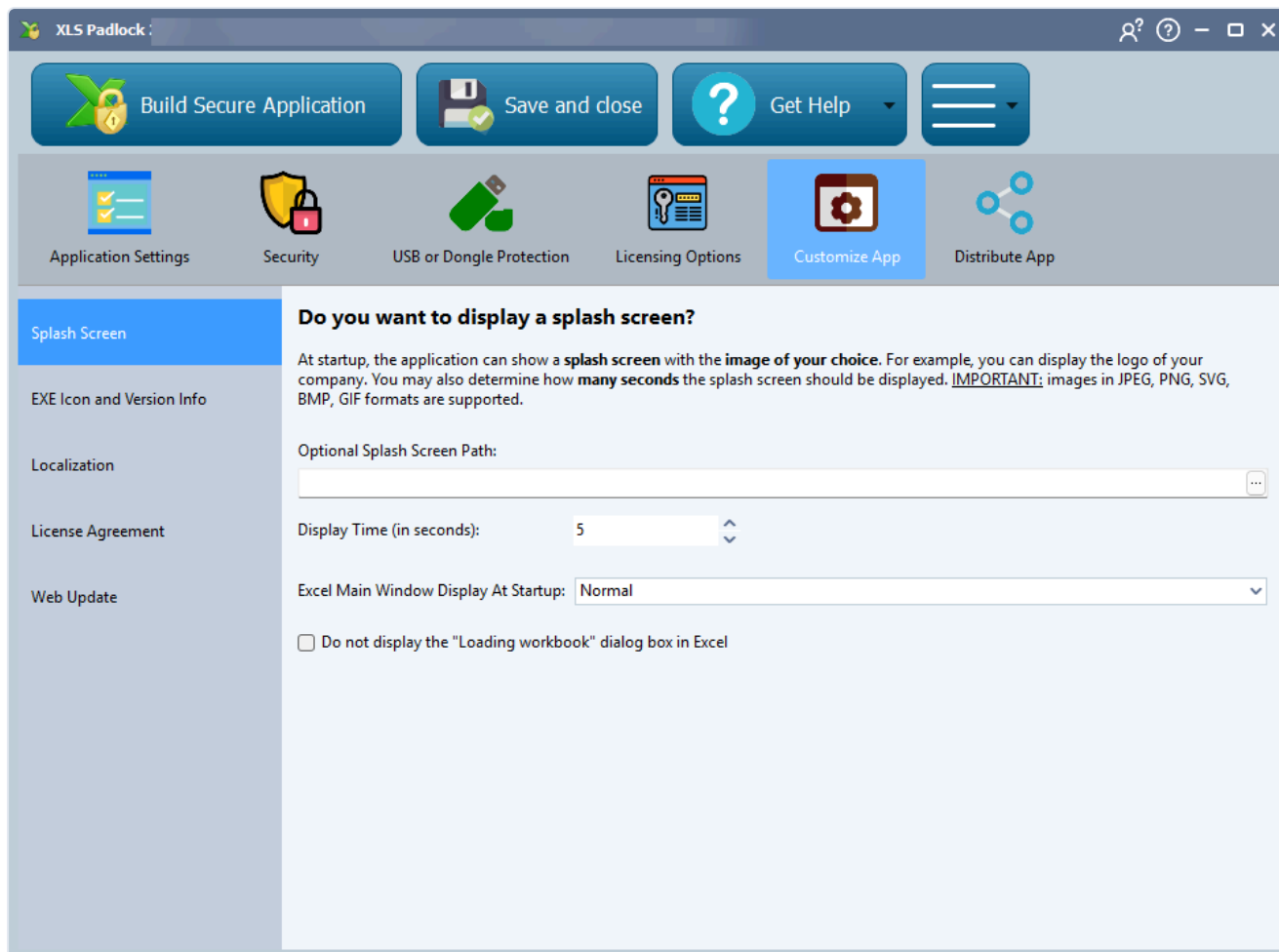
- L'interfaccia utente dell'applicazione è costruita con **UserForm**, non con fogli di calcolo.
- Non si desidera che gli utenti finali vedano, modifichino o interagiscano con il foglio di calcolo sottostante.
- Si desidera un aspetto pulito e personalizzato all'avvio, senza schermata iniziale di Excel, senza voce di Excel nella barra delle applicazioni.

Se l'applicazione si basa su fogli di calcolo visibili (griglie per l'inserimento dati, dashboard rese sotto forma di celle, ecc.), **non** attivare questa opzione, poiché i fogli di calcolo non saranno visibili agli utenti.

## Come attivarla

---

1. Aprire il progetto in XLS Padlock.
2. Accedere alla pagina **Splash Screen** (sotto Application Customization, personalizzazione dell'applicazione).
3. Spuntare l'opzione **"Run as a VBA-only app (Excel main window fully hidden)"** (esegui come app solo VBA, finestra principale di Excel completamente nascosta).



Questo è tutto per quanto riguarda XLS Padlock. Ora è necessario aggiungere un piccolo frammento di codice VBA alla cartella di lavoro affinché il proprio UserForm venga mostrato all'avvio dell'applicazione.

## Aggiungere il codice VBA di avvio

Nell'editor VBA di Excel, aprire il modulo `ThisWorkbook` e incollare:

```
Private Sub Workbook_Open()
 Application.Visible = False
 UserForm1.Show
End Sub
```

Sostituire `UserForm1` con il nome del form che si desidera visualizzare per primo.

### Perché entrambi, la casella di controllo \*e\* il codice VBA?

La casella di controllo di XLS Padlock garantisce che la finestra principale di Excel non diventi mai visibile **prima** dell'esecuzione del codice VBA (nessun lampeggiamento, nessuna icona nella barra delle applicazioni). La riga `Application.Visible = False` in `Workbook_Open` mantiene Excel nascosto **dopo** che Excel ha terminato l'inizializzazione: Excel riafferma internamente la propria visibilità durante l'avvio, quindi il VBA deve imporre la regola una volta che la cartella di lavoro è completamente caricata. L'utilizzo di entrambi i livelli offre un'esperienza utente pulita e priva di sfarfallio.

## Cosa vede l'utente finale

Con la casella di controllo attivata e il frammento VBA riportato sopra:

- **Nessuna schermata iniziale di Excel** all'avvio.
- **Nessuna finestra principale di Excel** in qualsiasi momento del ciclo di vita dell'applicazione.
- **Nessuna voce "Excel" nella barra delle applicazioni di Windows**, soltanto il proprio UserForm.
- L'applicazione si comporta come un programma Windows autonomo costruito attorno al proprio form.

Se è stata configurata anche una [splash screen](#) ed è stata nascosta la [finestra di dialogo Loading workbook](#), l'intera sequenza di avvio rimane all'interno dell'interfaccia personalizzata fin dal primo fotogramma.

## Chiusura dell'applicazione

Poiché Excel è nascosto, il pulsante di chiusura standard di Excel non è disponibile per gli utenti. Di norma si chiude l'applicazione dal proprio UserForm utilizzando:

```
Private Sub btnClose_Click()
 Application.Quit
End Sub
```

Oppure, se si desidera lasciare Excel in esecuzione ma scaricare la propria interfaccia:

```
Unload Me
```

## Interazione con altre impostazioni

Impostazione	Comportamento quando "VBA-only app" è attivata
<b>Excel Main Window Display at Startup</b> (Normal / Maximized / Minimized)	Ignorata, Excel si avvia sempre nascosto.

Impostazione	Comportamento quando "VBA-only app" è attivata
<b>Do not display the "Loading workbook" dialog box</b>	Si consiglia di attivarla anch'essa, per un avvio completamente silenzioso.
<b>Splash Screen</b>	Funziona come configurata. Visualizzata ancor prima del caricamento di Excel.

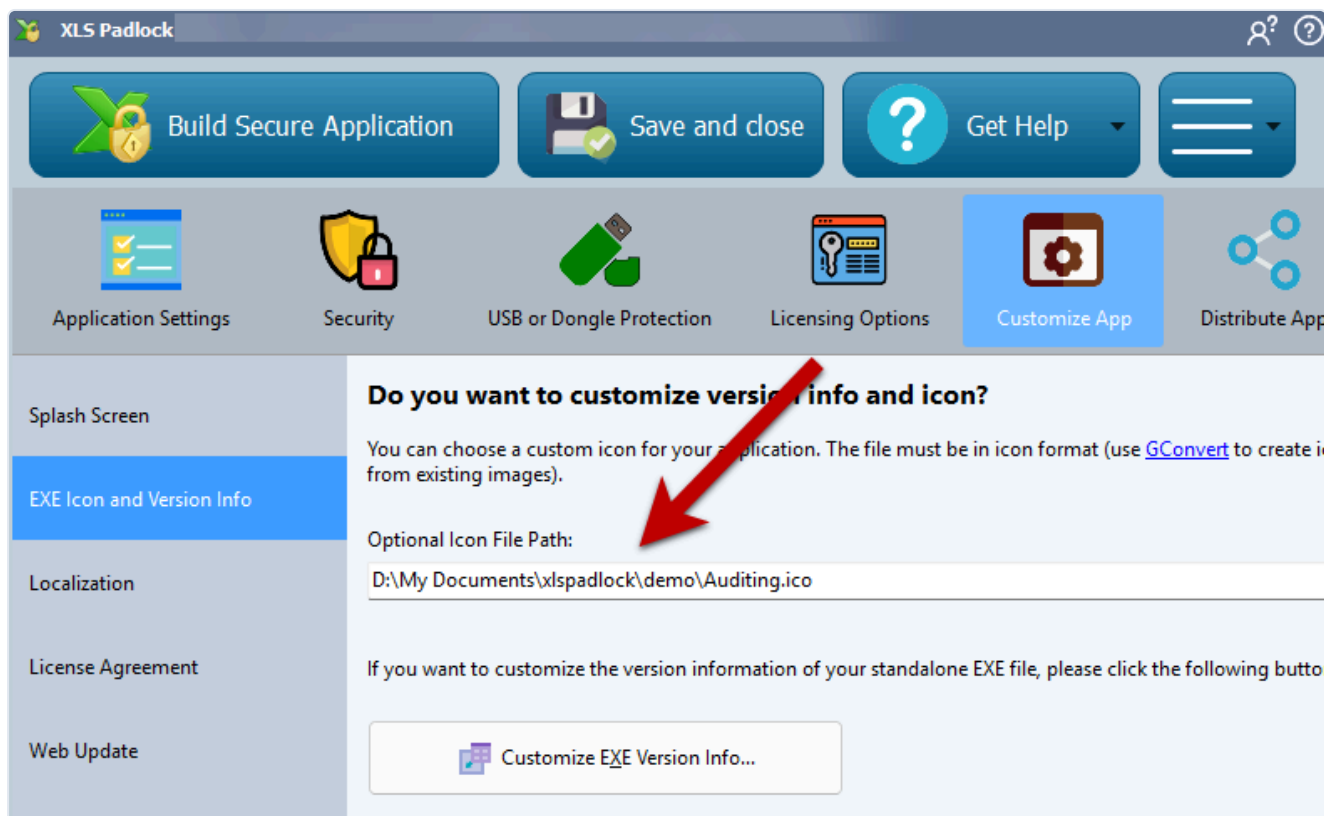
## Vedi anche

---

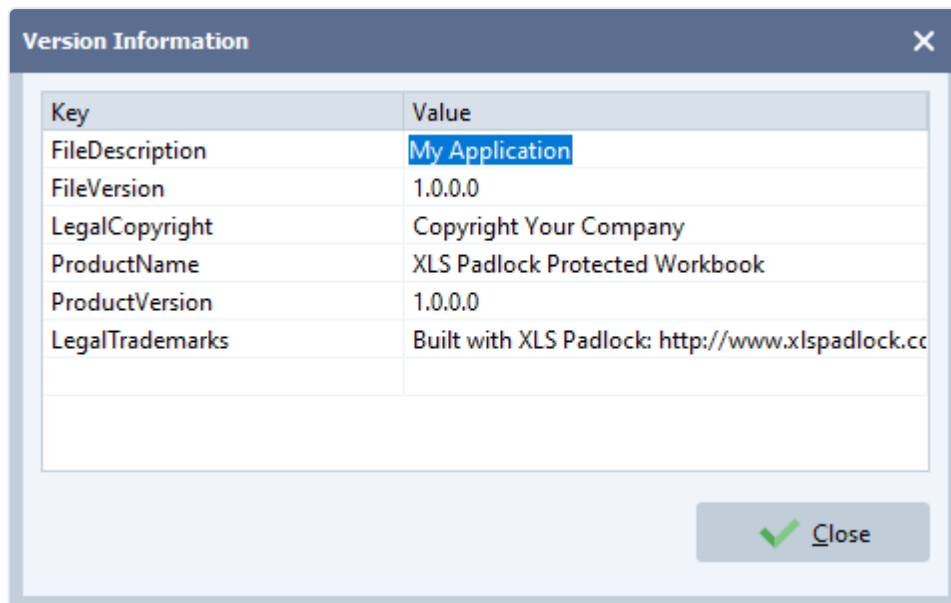
- [Come configurare la Splash Screen](#)
- [Come impostare la visualizzazione della finestra principale di Excel all'avvio](#)
- [Come nascondere la finestra di dialogo "Loading workbook"](#)
- [VBA API Cookbook](#)

# Informazioni sulla versione dell'EXE

Le **informazioni di versione** di un file eseguibile costituiscono una sezione di risorse speciale che contiene dettagli come il numero di versione del file, il sistema operativo previsto, il nome di file originale e le informazioni sul copyright. Questi dati vengono incorporati nell'EXE compilato. Quando sono inclusi, gli utenti possono consultarli facendo clic con il pulsante destro del mouse sull'icona del programma, selezionando "Proprietà" e accedendo alla scheda "Dettagli" (oppure premendo **Alt+Enter** sul file in Esplora risorse).



XLS Padlock consente di incorporare le proprie informazioni di versione nel file EXE autonomo. Fare clic su "**Customize EXE Version Info**" (Personalizza le informazioni di versione dell'EXE) per aprire una finestra con i seguenti campi:



- **File Description:** una breve descrizione del contenuto della propria applicazione.
- **Company Name:** il nome della propria azienda.
- **File Version:** il numero di rilascio del file `.exe`, nel formato `X.X.X.X` (ad esempio, `1.20.34.45`). Questo valore viene utilizzato anche dalla [funzione di aggiornamento Web](#) per verificare la presenza di nuove versioni dell'applicazione.
- **Legal Copyright:** la propria nota di copyright, ad esempio "Copyright © [annee] Your Company. All rights reserved."
- **Product Name:** il nome del proprio prodotto o applicazione, che generalmente coincide con il titolo dell'applicazione.
- **Product Version:** il numero di rilascio del proprio prodotto. Spesso corrisponde al File Version.
- **Legal Trademarks:** eventuali marchi registrati che si desidera includere.

#### Info

Il File Version viene utilizzato dalla [funzione di aggiornamento Web](#).

# Opzioni da riga di comando

Il file EXE dell'applicazione compilata supporta diverse **opzioni della riga di comando** (note anche come argomenti) che automatizzano azioni specifiche all'avvio. È possibile utilizzarle nei collegamenti o negli script per controllare il comportamento dell'applicazione al momento dell'avvio.

## SUGGERIMENTO

👉 Questi argomenti della riga di comando possono anche essere recuperati in fase di esecuzione utilizzando [l'API VBA per leggere i parametri della riga di comando](#).

## Opzioni disponibili

### -deact

Avvia il processo di disattivazione, consentendo a un utente di spostare la propria licenza su un altro computer.

```
MYAPP.EXE -deact
```

### -deL

Cancella tutti i file di salvataggio sicuri dell'applicazione e carica la cartella di lavoro originale. **Da utilizzare con cautela, poiché questa azione è irreversibile.**

```
MYAPP.EXE -deL
```

### -enterkey

Apri la finestra di dialogo di attivazione, consentendo all'utente di inserire una nuova chiave di attivazione. Questo è utile per sostituire una chiave vecchia o scaduta.

```
MYAPP.EXE -enterkey
```

### -load

Apri una finestra di dialogo di file che invita l'utente a selezionare un file di salvataggio da caricare.

```
MYAPP.EXE -load
```

### -reset

Annulla tutte le modifiche non salvate dell'ultima sessione e carica la cartella di lavoro originale, non modificata. Questo non elimina alcun file di salvataggio `.xlsc`.

```
MYAPP.EXE -reset
```

#### **-webupdate**

Forza l'applicazione a verificare immediatamente online la presenza di una nuova versione, in base alle impostazioni "Web Update".

```
MYAPP.EXE -webupdate
```

## Caricamento di un file di salvataggio specifico

---

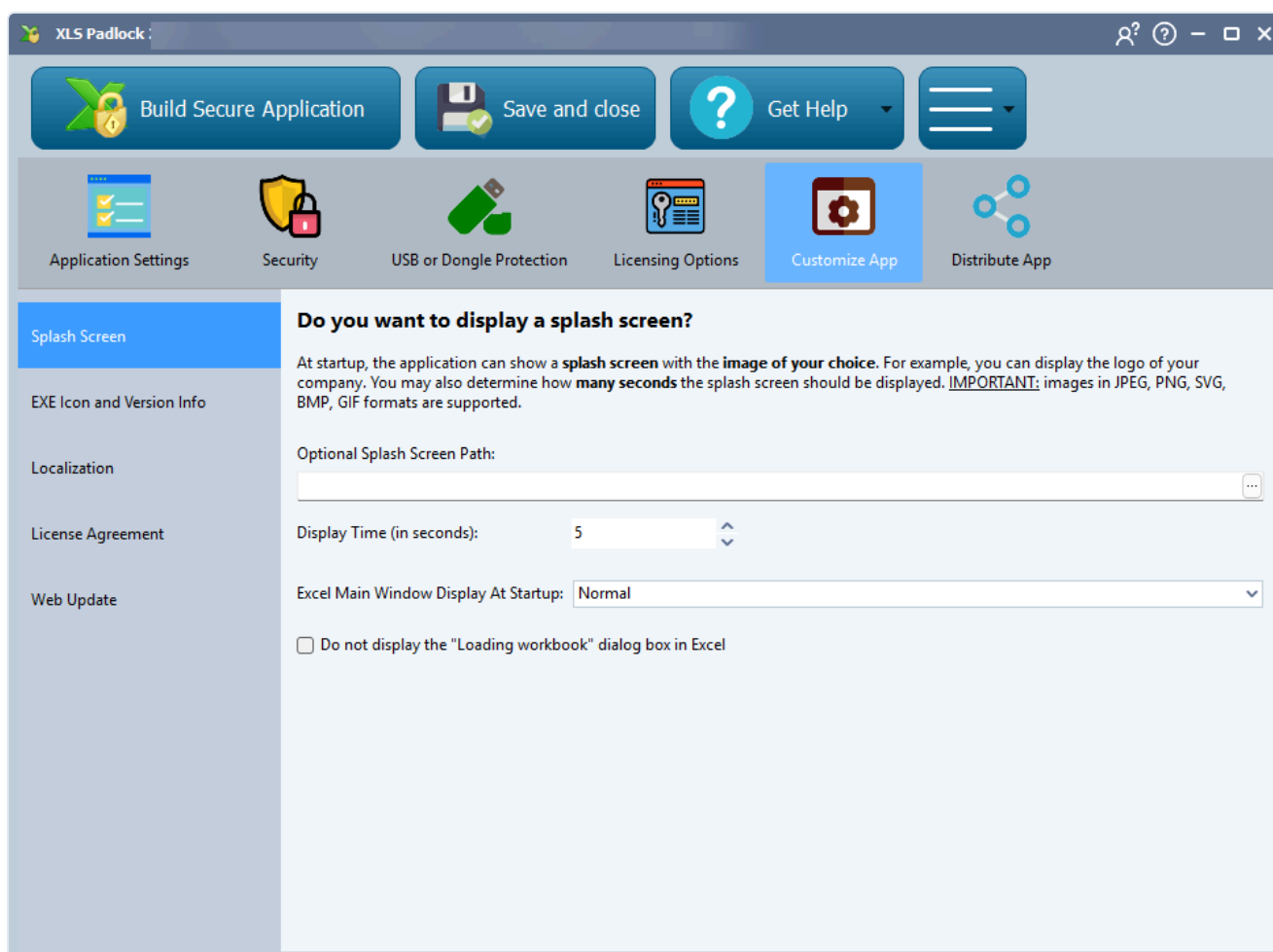
È inoltre possibile avviare l'applicazione passando come argomento il percorso completo di un file di salvataggio sicuro (`.xlsc` o `.xlscx`). Questo caricherà automaticamente il file specificato all'avvio, senza alcuna richiesta.

```
MyApp.exe "D:\My Documents\123.xlsc"
```

# Nascondere la finestra di caricamento

Per impostazione predefinita, all'avvio dell'applicazione protetta, Excel mostra una piccola finestra di dialogo con il messaggio **"Loading workbook, please wait..."**.

È possibile nascondere questa finestra di dialogo attivando l'opzione **"Do not display the "Loading workbook" dialog box in Excel"** (Non visualizzare la finestra di dialogo "Loading workbook" in Excel), situata nella pagina [Splash Screen](#). Ciò è utile per offrire un'esperienza di avvio più ordinata, in particolare quando si utilizza già uno splash screen personalizzato.



È inoltre possibile [nascondere questo messaggio in anticipo tramite codice VBA](#), il che offre un maggiore controllo sul momento in cui la finestra di dialogo scompare.

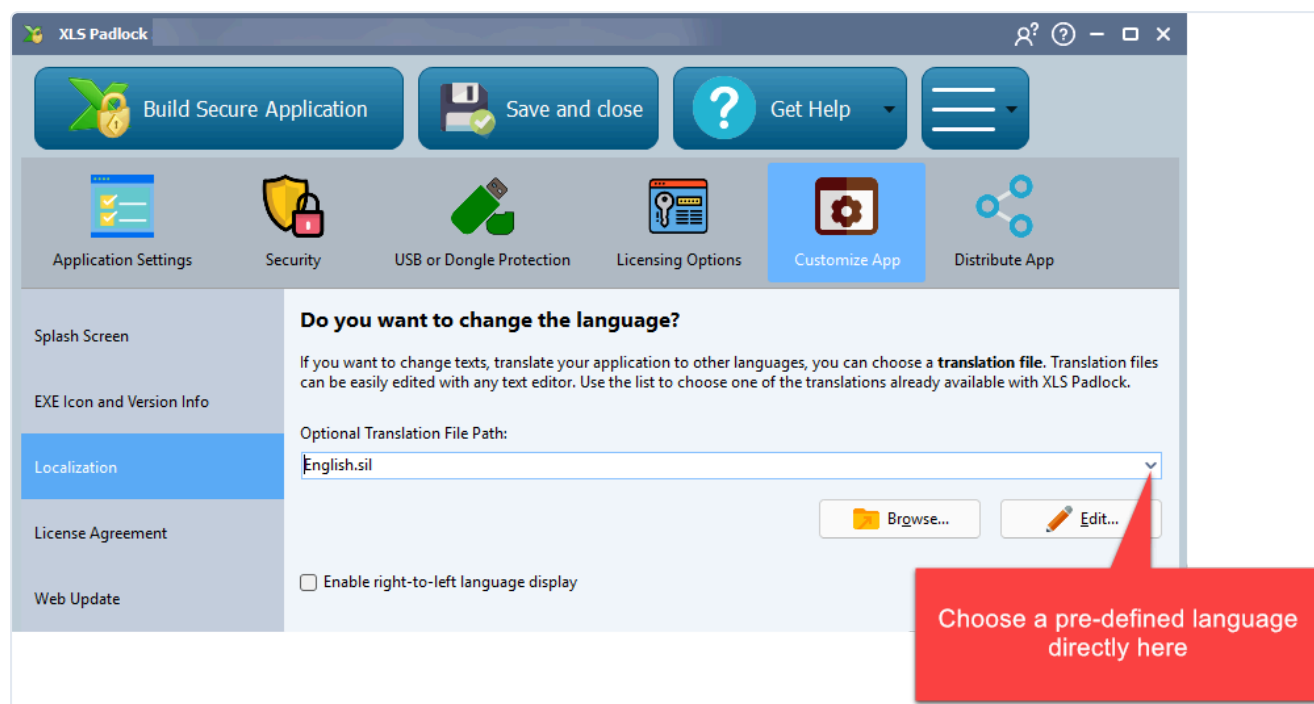
👉 Vedere anche

- [Come configurare lo Splash Screen](#)
- [Come impostare la visualizzazione della finestra principale di Excel all'avvio](#)

# Localizzazione e traduzione

XLS Padlock offre il **supporto alla localizzazione** per le cartelle di lavoro protette, consentendo di visualizzare tutto il testo dell'interfaccia (titoli delle finestre di dialogo, messaggi, richieste e avvisi di esecuzione) nella lingua desiderata.

È possibile configurare la localizzazione dell'applicazione della cartella di lavoro protetta qui:



## Formato del file di traduzione

A partire da XLS Padlock 2026, i file di traduzione utilizzano il formato **GNU gettext standard** con estensione `.po` (e il suo equivalente binario compilato `.mo`). È lo stesso formato utilizzato dalle distribuzioni Linux, Python, PHP, WordPress e da innumerevoli altre applicazioni, il che significa che è possibile modificare le traduzioni con uno dei numerosi strumenti gettext maturi disponibili, tra cui l'editor libero e multiplatforma [Poedit](#).

L'applicazione compilata incorpora una singola risorsa `.mo` al momento della compilazione, caricata interamente dalla memoria durante l'esecuzione: nessun file temporaneo viene estratto sul disco.

XLS Padlock viene fornito con file `.po` pre-tradotti per **10 lingue**, disponibili nella sottocartella `locale\` della directory di installazione di XLS Padlock:

File	Lingua
<code>en.po</code>	Inglese
<code>fr.po</code>	Francese
<code>es.po</code>	Spagnolo
<code>pt.po</code>	Portoghese (Brasile)

File	Lingua
nL.po	Olandese
de.po	Tedesco
ar.po	Arabo (layout da destra a sinistra supportato)
it.po	Italiano
zh_CN.po	Cinese semplificato
zh_TW.po	Cinese tradizionale

## Richiedere una nuova lingua

Se si necessita di una lingua che non figura nell'elenco, si prega di [contattarci](#). È inoltre possibile creare un proprio file `.po` partendo da uno di quelli sopra come punto di partenza. Vedere *Modifica dei file di traduzione* di seguito.

## Come definire una lingua per la cartella di lavoro protetta

- Nella pagina **Localization**, fare clic su **Browse** accanto al campo *Translation file*. La finestra di dialogo si apre per impostazione predefinita sulla cartella `locale\` dell'installazione di XLS Padlock, dove si trovano i 10 file `.po` pronti all'uso. Scegliere la lingua desiderata oppure passare a un file `.po` personalizzato.
- È inoltre possibile incollare manualmente il percorso completo nel campo *Translation file*, oppure lasciarlo vuoto per utilizzare la traduzione inglese predefinita incorporata in XLS Padlock.

### IMPORTANTE

Il file di traduzione selezionato deve essere accessibile al percorso specificato **quando si compila l'applicazione**. XLS Padlock legge e compila questo file durante il processo di compilazione e incorpora la risorsa `.mo` risultante direttamente nell'EXE protetto. Non esiste alcuna dipendenza a runtime dal file `.po` originale.

## Modifica dei file di traduzione

I file `.po` sono semplici file di testo UTF-8. È possibile modificarli in due modi:

- Con Poedit** ([poedit.net](https://poedit.net), gratuito, multiplatforma), la scelta consigliata. Mostra le stringhe sorgente e tradotte affiancate, segnala le voci approssimative (fuzzy) e non tradotte, convalida la sintassi al salvataggio e produce un output UTF-8 pulito.
- Con qualsiasi editor di testo** (Notepad++, VS Code, ...). Salvare in **UTF-8 senza BOM** e mantenere una coppia `msgid / msgstr` per blocco. Le righe che iniziano con `#` sono commenti e non vengono mostrate all'utente finale.

Si consiglia di **copiare uno dei file `.po` pre-fabbricati** in una nuova posizione (ad esempio accanto alla propria cartella di lavoro), modificare la copia e poi indirizzare XLS Padlock al nuovo file. La modifica dei

file all'interno della cartella di installazione di XLS Padlock richiede generalmente diritti di amministratore e potrebbe essere sovrascritta da un futuro aggiornamento.

## Migrazione dai vecchi file di traduzione `.sil`

Se è stata personalizzata l'interfaccia utente della cartella di lavoro protetta con XLS Padlock 2025.3 o versioni precedenti, la traduzione era memorizzata come file `.sil` (il vecchio formato di traduzione testuale utilizzato dalle versioni precedenti). XLS Padlock 2026 non legge più i file `.sil` al momento della compilazione. Il formato è stato sostituito da `.po`.

Per evitare di digitare nuovamente tutto, la distribuzione di XLS Padlock include un piccolo **assistente di migrazione SIL → PO** in `tools\sil_to_po\` nell'albero dei sorgenti:

```
python tools/sil_to_po/sil2po_customer.py your_translation.sil --lang fr --stats
```

Ciò produce `customer_fr.po` nella cartella corrente, automaticamente unito con il file `.po` canonico attuale in modo che:

- la colonna corrispondente alla lingua di destinazione venga riportata,
- le stringhe aggiunte dopo la stesura del file `.sil` appaiano con un `msgstr` vuoto (pronte da compilare in Poedit),
- le stringhe la cui sorgente inglese è leggermente cambiata da una versione all'altra siano contrassegnate `#, fuzzy` per la verifica,
- le stringhe che non esistono più in XLS Padlock siano aggiunte alla fine come voci obsolete `#~` (oppure eliminate del tutto con `--drop-orphans`).

Per migrare più lingue in una sola esecuzione, separarle con virgole:

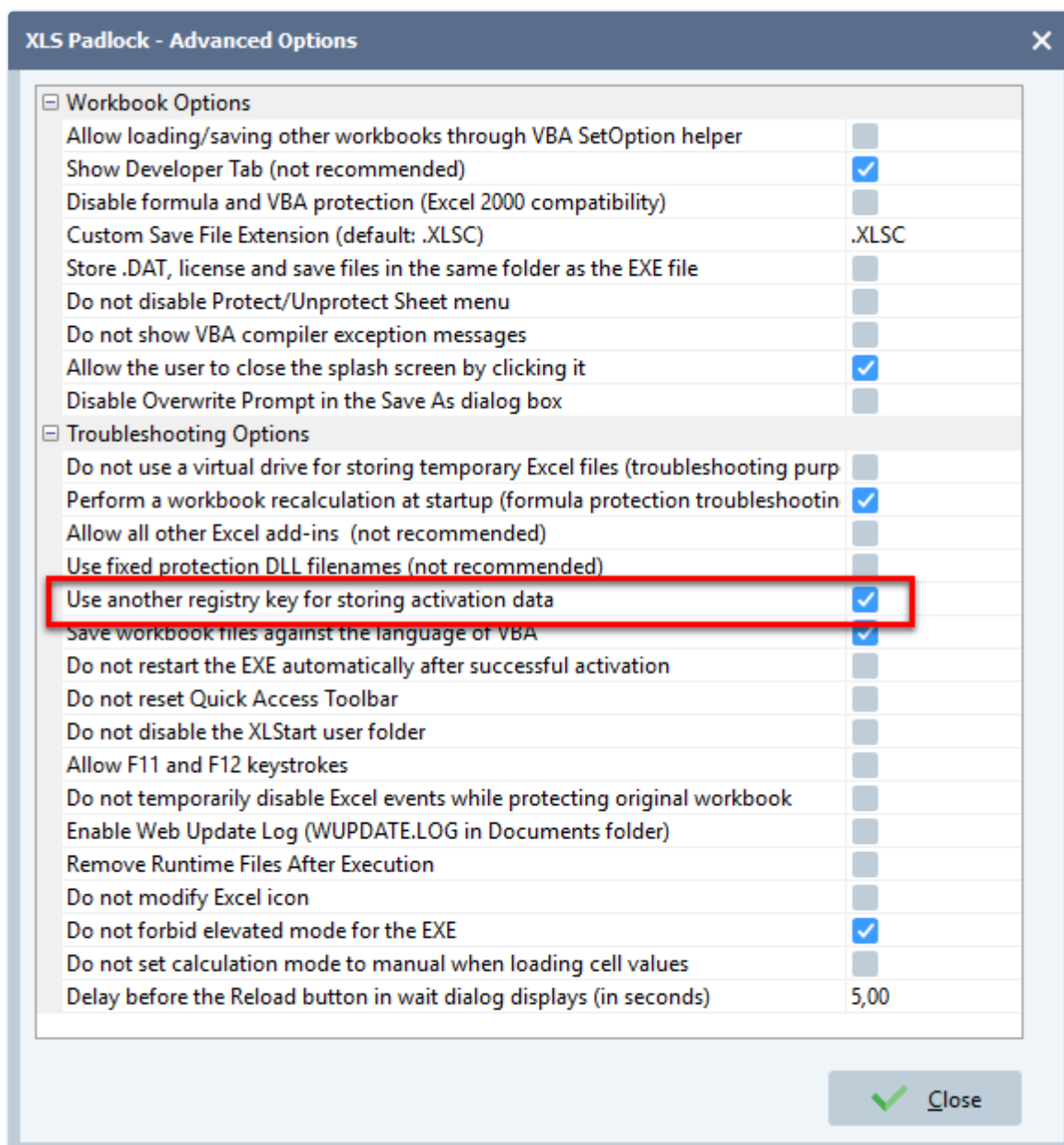
```
python tools/sil_to_po/sil2po_customer.py your_translation.sil --lang fr,es,de --stats
```

I file `.po` risultanti sono immediatamente utilizzabili dal selettore *Translation file* nel XLS Padlock Designer. Lo script assistente e il suo file complementare `README.md` (riferimento completo delle opzioni, algoritmo di corrispondenza, esempi dettagliati) sono forniti nella sottocartella `tools\sil_to_po\` della directory di installazione di XLS Padlock. Aprire `README.md` in qualsiasi editor di testo per leggere la documentazione completa in locale.

Per eseguire l'assistente è richiesto Python 3.8 o successivo. Nessuna dipendenza Python esterna: lo script è autonomo.

# Opzioni avanzate

In «Application Settings», fare clic sul pulsante «Configure Advanced Options» per visualizzare la finestra seguente:



## Riservato solo agli utenti avanzati

Queste opzioni sono destinate agli utenti avanzati. Modificarle solo se se ne comprende lo scopo o se consigliato dal nostro supporto tecnico.

## ## Opzioni avanzate disponibili

- **Allow loading/saving other workbooks through VBA SetOption helper** (consenti il caricamento/salvataggio di altre cartelle di lavoro tramite l'assistente VBA SetOption): un bypass di

sicurezza per operazioni VBA specifiche. Consultare [Caricamento/salvataggio di cartelle di lavoro tramite l'assistente VBA SetOption](#).

- **Show Developer Tab** (mostra la scheda Sviluppo): rende visibile la scheda Sviluppo di Excel (non consigliato per la sicurezza).
- **Disable formula and VBA protection** (disattiva la protezione delle formule e del VBA): per la compatibilità con Excel 2000.
- **Custom Save File Extension** (estensione personalizzata dei file di salvataggio): modifica l'estensione `.XLSC` predefinita dei file di salvataggio.
- **Store .DAT, license and save files in the same folder as the EXE file** (memorizza i file .DAT, di licenza e di salvataggio nella stessa cartella del file EXE): attiva la modalità portatile.
- **Do not disable Protect/Unprotect Sheet menu** (non disattivare il menu Proteggi/Rimuovi protezione foglio): mantiene attivati gli elementi di menu per la protezione dei fogli.
- **Do not show VBA compiler exception messages** (non mostrare i messaggi di eccezione del VBA Compiler): nasconde i messaggi di errore dettagliati del VBA Compiler. Consultare [Disattivare le informazioni di debug](#).
- **Allow the user to close the splash screen by clicking it** (consenti all'utente di chiudere la schermata iniziale facendo clic su di essa): consente agli utenti di saltare la schermata iniziale.
- **Disable Overwrite Prompt in the Save As dialog box** (disattiva la richiesta di sovrascrittura nella finestra di dialogo Salva con nome): impedisce la comparsa del messaggio «do you want to overwrite?».
- **Do not use a virtual drive for storing temporary Excel files** (non utilizzare un'unità virtuale per memorizzare i file temporanei di Excel): a scopo di risoluzione dei problemi.
- **Perform a workbook recalculation at startup** (esegui un ricalcolo della cartella di lavoro all'avvio): per la risoluzione dei problemi relativi alla protezione delle formule.
- **Allow all other Excel add-ins** (consenti tutti gli altri componenti aggiuntivi di Excel): abbassa la sicurezza attivando tutti i componenti aggiuntivi (non consigliato).
- **Use fixed protection DLL filenames** (utilizza nomi di file DLL di protezione fissi): non consigliato per la sicurezza.
- **Use another registry key for storing activation data** (utilizza un'altra chiave di registro per memorizzare i dati di attivazione): una soluzione alternativa per i conflitti di registro.
- **Save workbook files against the language of VBA** (salva i file della cartella di lavoro in base alla lingua del VBA): risolve alcuni problemi di localizzazione.
- **Do not restart the EXE automatically after successful activation** (non riavviare automaticamente l'EXE dopo un'attivazione riuscita): impedisce il riavvio automatico.
- **Do not reset Quick Access Toolbar** (non reimpostare la barra di accesso rapido): conserva le personalizzazioni dell'utente apportate alla barra di accesso rapido (QAT).
- **Do not disable the XLStart user folder** (non disattivare la cartella utente XLStart): consente l'esecuzione dei componenti aggiuntivi presenti nella cartella XLStart dell'utente.
- **Allow F11 and F12 keystrokes** (consenti i tasti F11 e F12): attiva questi tasti funzione.
- **Do not temporarily disable Excel events while protecting original workbook** (non disattivare temporaneamente gli eventi di Excel durante la protezione della cartella di lavoro originale): a scopo di

risoluzione dei problemi.

- **Enable Web Update Log** (attiva il registro degli aggiornamenti Web): crea un file `WUPDATE.LOG` nella cartella Documenti dell'utente per il debug della funzionalità di aggiornamento Web.
- **Remove Runtime Files After Execution** (rimuovi i file di runtime dopo l'esecuzione): pulisce i file temporanei.
- **Do not modify Excel icon** (non modificare l'icona di Excel): impedisce a XLS Padlock di modificare l'icona della finestra di Excel.
- **Do not forbid elevated mode for the EXE** (non vietare la modalità con privilegi elevati per l'EXE): consente all'applicazione di essere eseguita come amministratore.
- **Do not set calculation mode to manual when loading cell values** (non impostare la modalità di calcolo su manuale durante il caricamento dei valori delle celle): a scopo di risoluzione dei problemi.
- **Delay before the Reload button in wait dialog displays (in seconds)** (ritardo prima della comparsa del pulsante Ricarica nella finestra di dialogo di attesa, in secondi): personalizza la finestra di dialogo di attesa.

# Ignorare gli errori di elaborazione

---

L'opzione "Ignore errors when processing the workbook (internal protection)" (Ignora gli errori durante l'elaborazione della cartella di lavoro, protezione interna) consente al processo di compilazione di proseguire anche se si verificano errori durante l'apertura della cartella di lavoro Excel. Questa funzionalità è concepita per semplificare il flusso di protezione, sopprimendo i messaggi di errore che altrimenti potrebbero interrompere il processo.

Quando questa opzione è attivata, XLS Padlock non si arresta e non visualizza richieste di errore se incontra problemi durante la fase di apertura della cartella di lavoro.

## Questa opzione non corregge gli errori

L'attivazione di questa opzione non risolve alcun problema sottostante della cartella di lavoro; si limita ad aggirarlo durante la compilazione. Dopo aver protetto la cartella di lavoro con questa opzione attivata, è necessario collaudarla a fondo per assicurarsi che tutte le funzionalità operino come previsto.

Se si riscontrano ancora errori nella cartella di lavoro compilata, valutare l'attivazione dell'opzione alternativa "[Use Excel automation for formula protection](#chapter-formula-protection-method)" (Usa l'automazione di Excel per la protezione delle formule), che può attenuare i problemi non risolti dal metodo standard di protezione interna.

# Disattivare le informazioni di debug

---

Per impostazione predefinita, se il VBA compiler rileva un errore nel codice compilato in fase di esecuzione, visualizza un messaggio di errore dettagliato per aiutare a eseguire il debug del problema.

Se non si desidera mostrare queste informazioni agli utenti finali, è possibile disattivarle. Accedere a [Configure Advanced Options](#) e disattivare l'opzione: **“Do not show VBA compiler exception messages”**.

# Interfaccia personalizzata di Excel

---

Come spiegato nella documentazione di Microsoft, è possibile personalizzare l'interfaccia utente di Excel collocando un file `.officeUI` appositamente predisposto in una cartella utente specifica.

XLS Padlock consente di specificare il proprio Excel.officeUI per l'applicazione protetta. Sostituirà automaticamente l'interfaccia predefinita dell'utente (se disponibile).

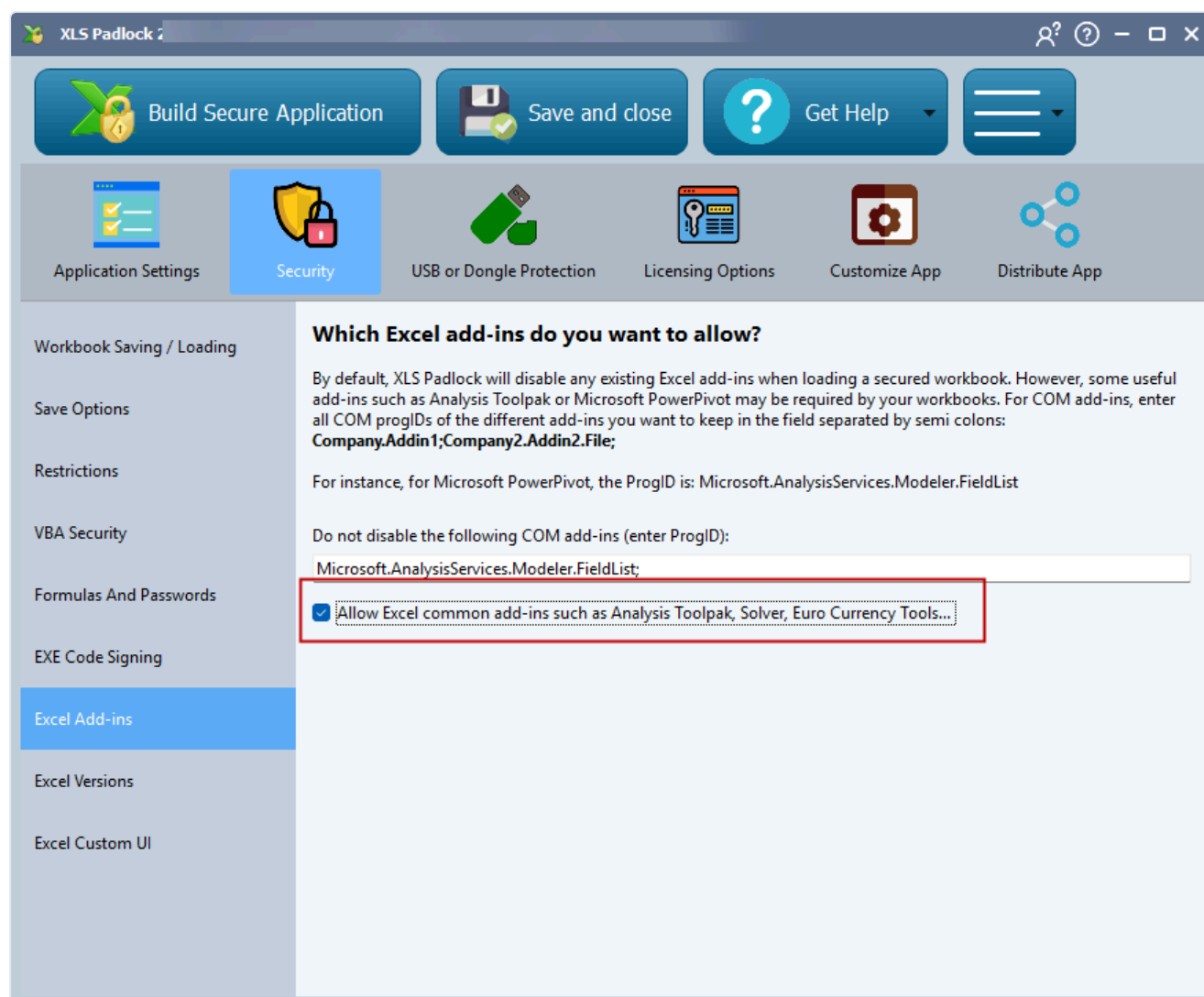
È sufficiente fornire il percorso completo del proprio file `.officeUI`. XLS Padlock lo compilerà direttamente nell'EXE finale, quindi non è necessario distribuire il file `.officeUI` separatamente.

Per maggiori informazioni sulla creazione di file XML di interfaccia personalizzata, consultare la [documentazione ufficiale di Microsoft Office](#).

# Componenti aggiuntivi di Excel

Per impostazione predefinita, XLS Padlock disattiva la maggior parte dei componenti aggiuntivi di Excel quando viene caricata una cartella di lavoro protetta. Si tratta di una misura di sicurezza volta a creare un ambiente controllato per la propria applicazione.

Tuttavia, se la cartella di lavoro dipende da componenti aggiuntivi specifici per funzionare correttamente, è possibile riattivarli in modo selettivo. Questa pagina spiega come autorizzare sia i componenti aggiuntivi COM sia i comuni componenti aggiuntivi integrati di Excel.



## Mantenere attivi componenti aggiuntivi COM specifici

Se la cartella di lavoro richiede un componente aggiuntivo COM (come Microsoft PowerPivot), è necessario specificarne il `ProgID` per impedire a XLS Padlock di disattivarlo.

Nel campo "Do not disable the following COM add-ins" (non disattivare i seguenti componenti aggiuntivi COM), inserire il `ProgID` di ciascun componente aggiuntivo che si desidera mantenere attivo. Se ve ne sono più di uno, separarli con punti e virgola ( ; ).

**Esempio:** per mantenere attivo Microsoft PowerPivot, occorre inserire i relativi ProgID:

```
Microsoft.AnalysisServices.Modeler.FieldList;PowerPivotExcelClientAddIn.NativeEntry.1
```

### Trovare un ProgID

Il ProgID (identificatore programmatico) è un nome univoco memorizzato nel registro di sistema di Windows per un componente COM. È spesso possibile trovarlo nella documentazione del componente aggiuntivo o ispezionando il registro di sistema. Microsoft fornisce ulteriori informazioni su questo argomento nel proprio sito web di assistenza.

## Autorizzare i componenti aggiuntivi comuni di Excel

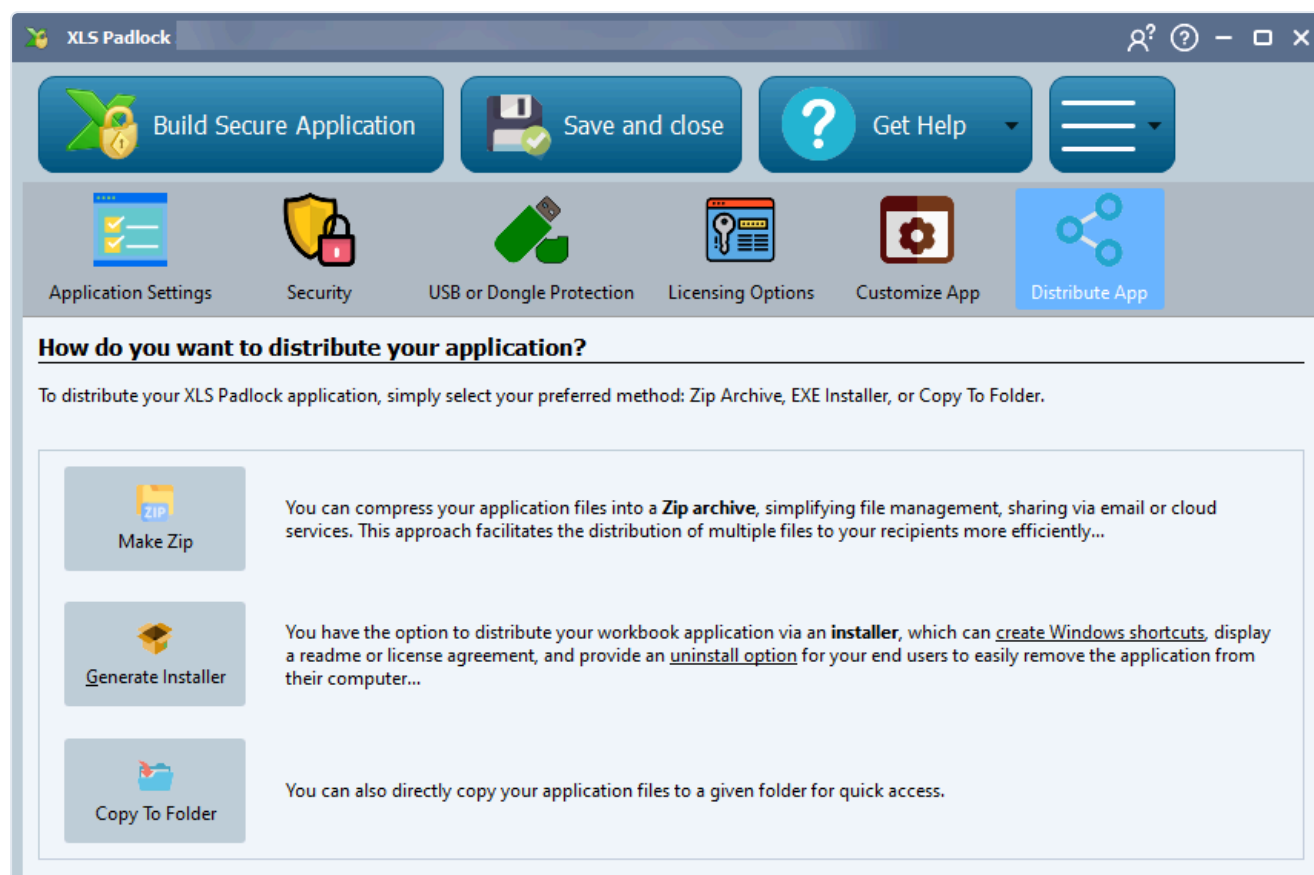
Excel include anche diversi componenti aggiuntivi integrati molto diffusi, come **Analysis ToolPak**, **Solver** e **Euro Currency Tools**.

Se la cartella di lavoro utilizza funzionalità di uno qualsiasi di questi, attivare l'opzione "**Allow Excel common add-ins...**" (consenti i componenti aggiuntivi comuni di Excel). In questo modo si garantirà il loro caricamento e la loro disponibilità all'avvio dell'applicazione protetta.

# Distribuire l'applicazione

Dopo aver [convertito la cartella di lavoro in un eseguibile \(.EXE\)](#), il passaggio successivo consiste nel distribuirla ai propri utenti.

XLS Padlock offre diversi metodi per semplificare questo processo: creare un archivio Zip, generare un programma di installazione EXE, oppure copiare direttamente i file in una cartella.



## Archivio Zip

Un archivio Zip è un metodo diffuso e semplice per distribuire un software. Comprimerne l'applicazione in un unico file `.zip` semplifica la gestione dei file per gli utenti. Questo metodo è particolarmente utile per condividere l'applicazione via e-mail o tramite servizi cloud, poiché riduce le dimensioni del file e raggruppa tutti i componenti necessari in un unico pacchetto.

Per creare un archivio Zip dell'applicazione, fare clic su **Make Zip** (Crea Zip) e scegliere la posizione in cui salvare il file `.zip`.

## Programma di installazione EXE

Per un tocco più professionale, è consigliabile distribuire l'applicazione con un programma di installazione EXE. Questo metodo migliora l'esperienza dell'utente offrendo un processo di installazione familiare,

comune negli ambienti Windows. Un programma di installazione può creare collegamenti sul Desktop e nel menu Start, mostrare un file readme o un contratto di licenza e fornire un programma di disinstallazione, consentendo agli utenti di rimuovere facilmente l'applicazione dal proprio computer.

➡ Per istruzioni dettagliate, consultare la nostra guida su [come creare un programma di installazione per la propria applicazione](#).

## Copiare in una cartella

---

Infine, copiare direttamente i file dell'applicazione in una cartella è il metodo di distribuzione più immediato. Comporta due semplici passaggi:

- Copiare tutti i file necessari dell'applicazione in una cartella.
- Condividere questa cartella con i propri utenti tramite un supporto fisico (come una chiavetta USB) o tramite trasferimento in rete.

Pur essendo meno rifinito rispetto agli altri metodi, questo approccio è il modo più rapido per far funzionare l'applicazione su un altro sistema, poiché non richiede alcun'utilità di compressione né software di installazione particolare.

👉 Vedere anche:

- [Come firmare digitalmente la propria applicazione](#)

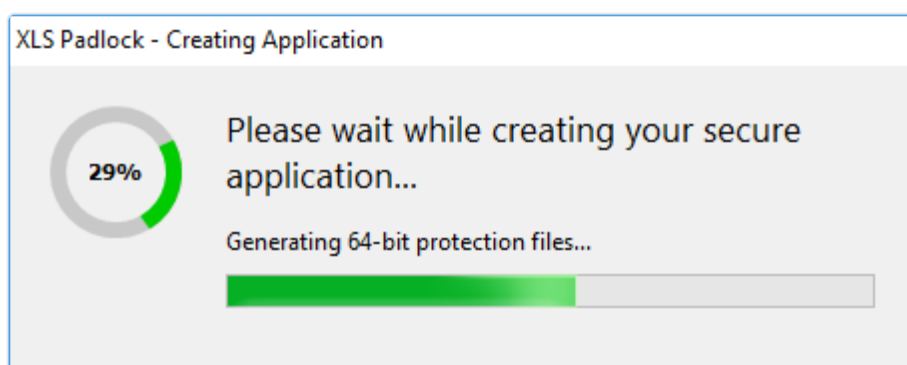
# Distribuire una cartella di lavoro protetta

## Compilare la cartella di lavoro Excel in EXE

In XLS Padlock, quando si fa clic su “**Build Secure Application**”



, viene visualizzata la finestra di dialogo di avanzamento della compilazione:



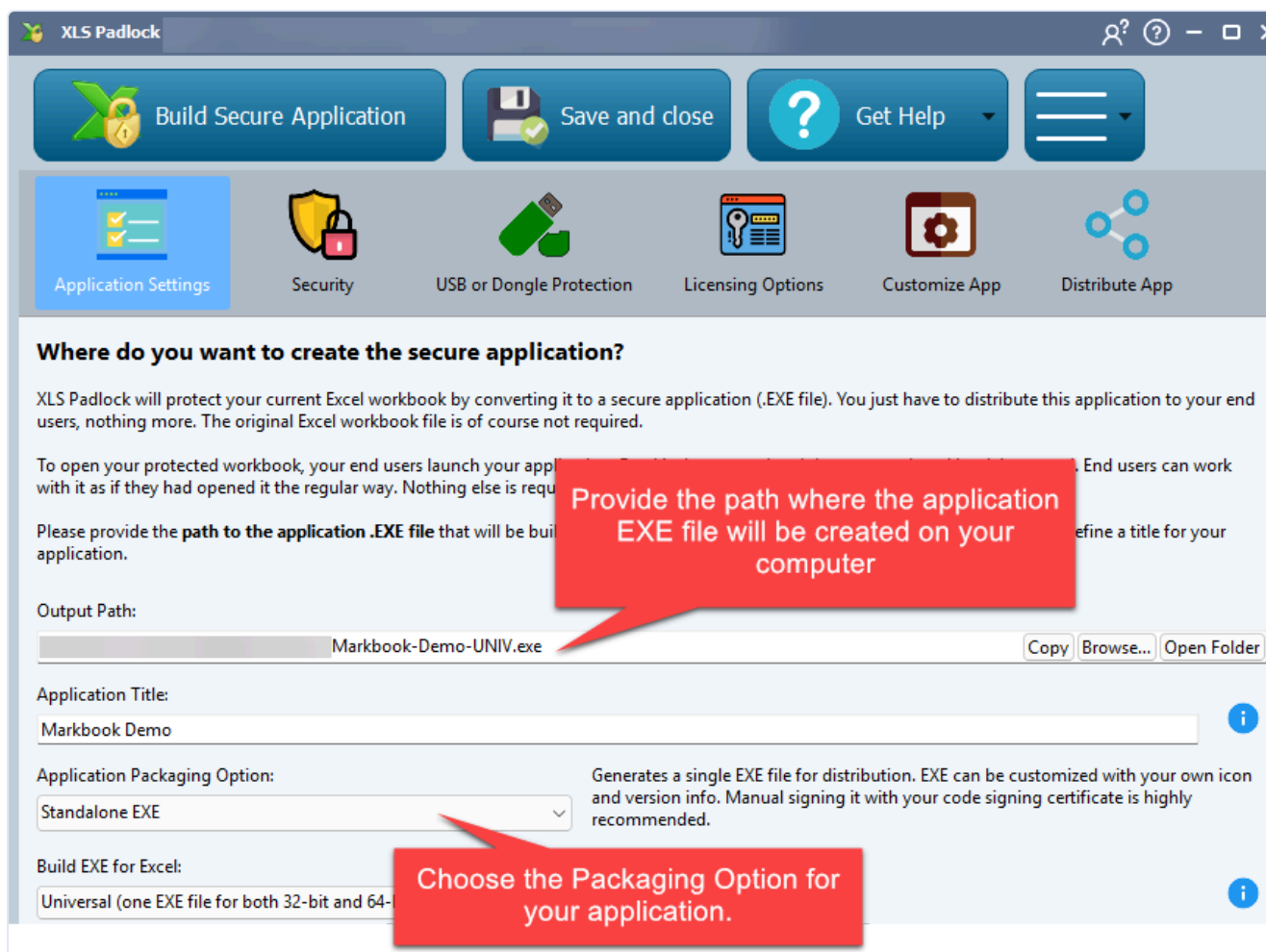
Al termine, **la propria applicazione sicura è pronta**. XLS Padlock ha generato un **file eseguibile** (.EXE) a partire dalla cartella di lavoro Excel.

**► D'ora in avanti, per accedere alla cartella di lavoro protetta, gli utenti finali avviano questo file .EXE. Una copia locale di Microsoft Excel è l'unico altro requisito.** Il file della cartella di lavoro originale **NON è PIÙ** necessario.

- All'avvio del file .EXE, Excel si avvia e la cartella di lavoro protetta si apre. Gli utenti finali possono interagire con essa come se l'avessero aperta nel modo consueto.
- Alcune funzionalità di Excel come “New Workbook”, Open e Save (facoltativamente) sono disabilitate per motivi di sicurezza.
- Se Excel non viene trovato sul computer dell'utente, viene visualizzato un messaggio di errore. È possibile configurare l'applicazione in modo da [richiedere versioni specifiche di Excel](#).

## Opzioni di pacchettizzazione dell'applicazione

XLS Padlock offre due formati per l'applicazione sicura, il che influisce sul modo in cui la si distribuisce e si gestiscono gli avvisi di sicurezza come Windows SmartScreen.



## EXE autonomo

XLS Padlock crea un singolo file eseguibile (.EXE) a partire dalla cartella di lavoro. È sufficiente distribuire questo singolo file agli utenti finali. La cartella di lavoro Excel originale non è necessaria.

### La firma del codice è altamente consigliata

Si consiglia vivamente di [firmare digitalmente i propri file EXE](#). La distribuzione di un file EXE non firmato causerà probabilmente la visualizzazione da parte di Windows SmartScreen di un avviso "Unknown Application". Sebbene l'EXE venga comunque eseguito, una firma digitale offre fiducia e una migliore esperienza utente.

### ### Bundle dell'applicazione EXE + XPLAPP

Questa opzione genera un file EXE, un file di accompagnamento `.bin64` e un file di dati `.xplapp` separato. L'EXE principale è pre-firmato dalla nostra società, il che ne facilita il riconoscimento da parte di Windows SmartScreen e dei software antivirus.

- È necessario distribuire tutti e tre i file insieme ( `.exe` , `.bin64` e `.xplapp` ).
- XLS Padlock può raggruppare questi file in un [singolo archivio Zip](#) o in un [programma di installazione](#) per facilitarne la distribuzione.

- Questa opzione riduce notevolmente il rischio di avvisi "Unknown Application" e di falsi positivi antivirus.

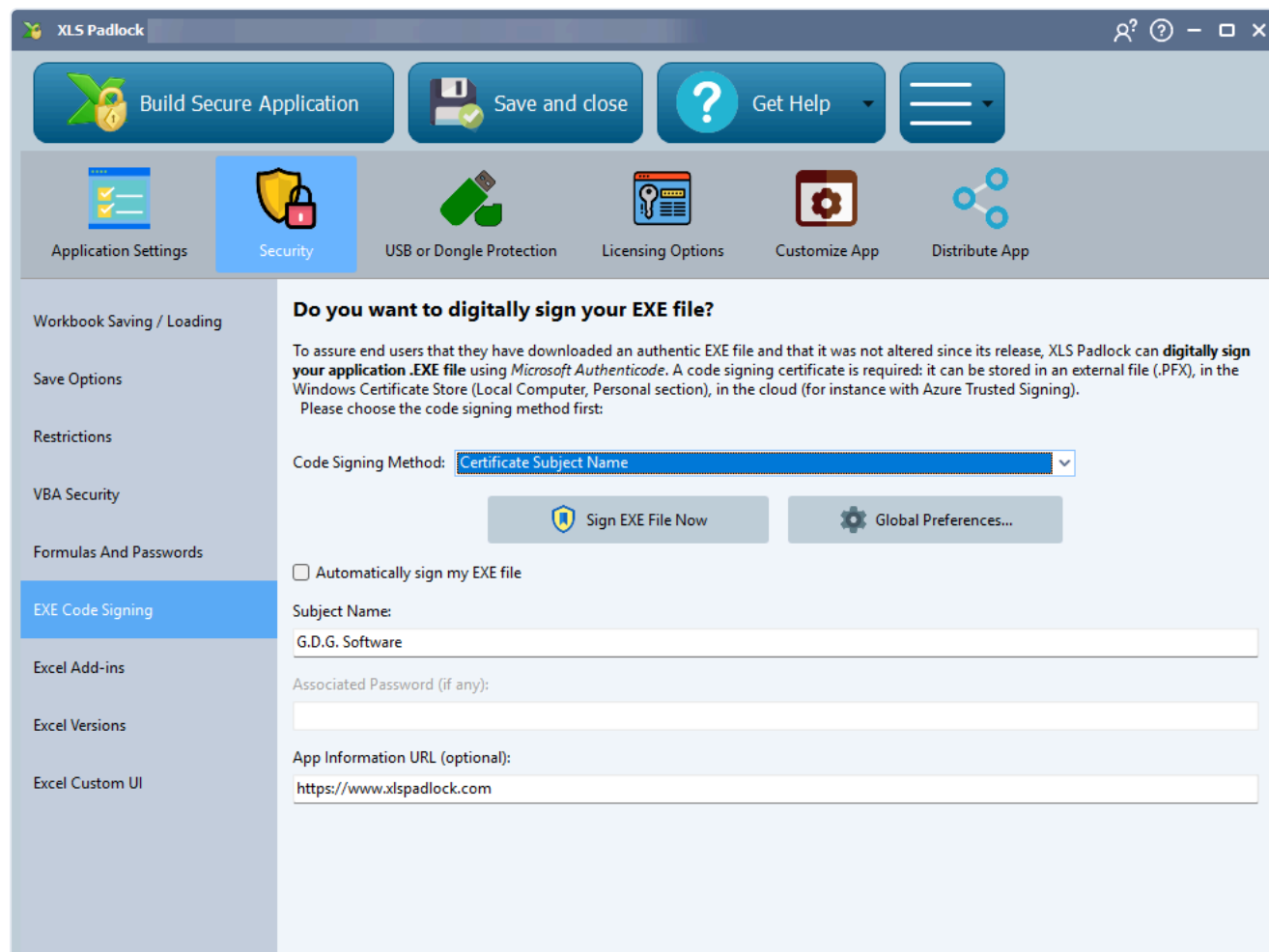
## Raccomandazioni per la distribuzione

---

1. **Firmare digitalmente i propri file EXE:** è il passaggio più importante per evitare i falsi positivi antivirus e instaurare fiducia presso i propri utenti. XLS Padlock può [automatizzare questo processo](#) se si dispone di un certificato di firma del codice.
2. **Disabilitare temporaneamente il proprio antivirus:** se si dispone di un programma antivirus attivo, valutare di disabilitarlo prima di compilare le cartelle di lavoro. I software antivirus possono talvolta segnalare erroneamente la creazione di nuovi file EXE sul computer.
3. **Non inviare i file EXE tramite e-mail:** la maggior parte dei provider di posta elettronica blocca i file `.exe`. È preferibile caricare i file della propria applicazione su un servizio di hosting di file come Dropbox, Google Drive o il proprio server web, e condividere il link di download. È inoltre possibile [pacchettizzare la propria applicazione in un archivio Zip](#).

# Firmare digitalmente l'EXE

Quando si **firma digitalmente il file EXE della cartella di lavoro autonoma**, si garantisce agli utenti finali che il file è autentico e non è stato manomesso. Questo processo, noto anche come firma del codice, utilizza la tecnologia Microsoft Authenticode® per verificare che il codice provenga da un editore attendibile.



XLS Padlock semplifica il processo di firma del codice gestendo internamente i passaggi necessari.

## Perché la firma del codice è importante

Se si prevede di distribuire le proprie applicazioni su Internet, la firma del codice è fortemente consigliata. Aiuta a evitare che i browser web e Windows mostrino avvisi "Unidentified Publisher" e può inoltre ridurre la probabilità di falsi positivi da parte dei software antivirus.

![Avviso di sicurezza di Windows che mostra un editore non identificato per un file EXE non firmato] (file:///D:/websites/xlspadlock/astro/src/content/docs/it/doc/code-sign-your-exe-file-digital-signature/images/image\_97.png)

## Come ottenere un certificato di firma del codice

Per firmare l'applicazione, è necessario un **Code Signing Certificate** valido rilasciato da un'autorità di certificazione (CA) attendibile come Sectigo o Digicert. Altri tipi di certificato, come SSL/TLS, non sono compatibili.

### Certificati basati su cloud

Un'alternativa sempre più diffusa ed economica è **Azure Trusted Signing**, un servizio basato su cloud di Microsoft. Elimina la necessità di token USB fisici e semplifica il processo di gestione. XLS Padlock supporta pienamente la [firma con Azure Trusted Signing](#).

### ## Certificati basati su token (HSM)

Dal 1° giugno 2023, tutte le nuove chiavi private dei certificati di firma del codice devono essere archiviate su hardware sicuro, come un token USB conforme allo standard FIPS 140-2 Livello 2 o un modulo di sicurezza hardware (HSM). Questo rafforza la sicurezza impedendo il furto delle chiavi. XLS Padlock funziona perfettamente con i certificati basati su token; assicurarsi semplicemente che il token sia collegato al computer quando si genera l'applicazione.

## Configurare la firma del codice in XLS Padlock

Nell'interfaccia di XLS Padlock, accedere alla scheda **Security -> EXE Code Signing**. Per abilitare la firma, scegliere il **Code Signing Method** preferito:

- **PFX File:** utilizza un certificato archiviato in un file `.pfx`. È un metodo legacy per i certificati più vecchi.
- **Certificate Subject Name:** individua il certificato nell'archivio certificati di Windows tramite il suo Subject Name (nome del soggetto). È un metodo comune per i certificati su token hardware.
- **Certificate Thumbprint:** individua il certificato nell'archivio certificati di Windows tramite il suo Thumbprint univoco (impronta, un hash SHA-1). È spesso il metodo più affidabile.
- **SignTool Commands:** un metodo avanzato che consente di fornire comandi personalizzati per l'utilità `SignTool.exe` di Microsoft, offrendo la massima flessibilità.
- **Azure Trusted Signing:** firma l'applicazione con il servizio basato su cloud di Microsoft. Consultare il nostro [tutorial su Azure Trusted Signing](#) per maggiori dettagli.

Azure CLI richiesta

Per utilizzare questo metodo, è necessario prima installare la Microsoft Azure CLI ed effettuare l'accesso tramite `az login`.

## Eseguire la firma

- **Firma manuale:** fare clic su **Sign EXE File Now** per firmare immediatamente l'ultimo file EXE generato.

- **Firma automatica:** selezionare **Automatically sign my EXE file** affinché XLS Padlock firmi l'EXE ogni volta che si genera l'applicazione.

#### Risoluzione dei problemi

Se si verifica un errore di firma, controllare i messaggi dettagliati nel log di compilazione di XLS Padlock. Questo file di log è solitamente denominato `[Your Workbook Filename].xplcompil.log` e si trova nella stessa directory della cartella di lavoro.

# Creare un programma di installazione

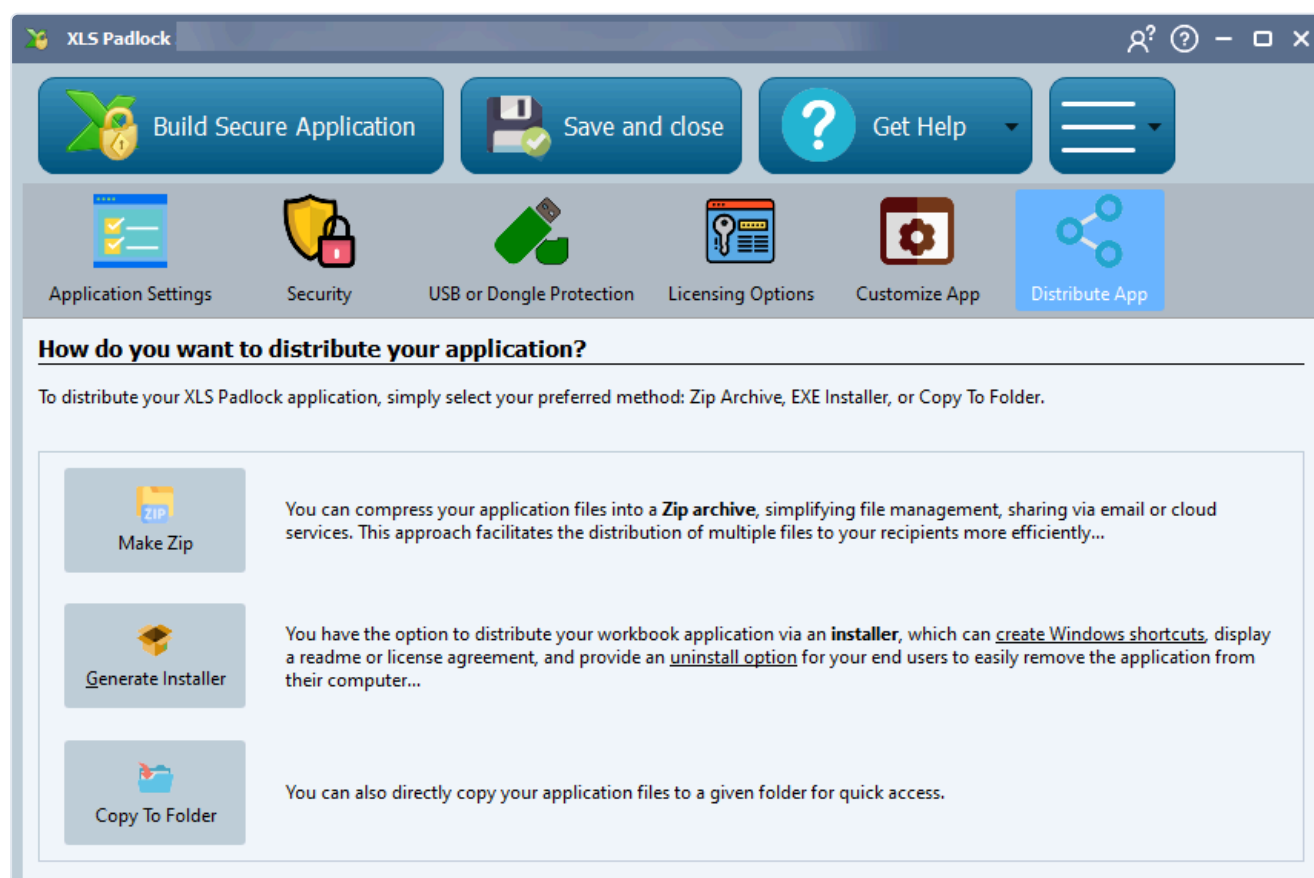
Impacchettare la cartella di lavoro compilata in un programma di installazione professionale (o programma di setup) è un modo eccellente per distribuire l'applicazione. I programmi di installazione offrono un'esperienza utente familiare e possono gestire attività come l'installazione di più file (ad esempio un file readme), la visualizzazione di un contratto di licenza, la creazione di collegamenti e la fornitura di un programma di disinstallazione per una rimozione pulita.

XLS Padlock si integra con il nostro software, [Paquet Builder](#), per generare programmi di installazione personalizzati e compatti. **Paquet Builder deve essere installato sul computer per poter utilizzare questa funzionalità.**

*Paquet Builder è uno strumento potente che combina un creatore di archivi auto-estraenti 7-Zip con un generatore di routine di setup. Consente di creare programmi di installazione flessibili e compatti per una distribuzione software professionale, impacchettare documenti e file di programma, realizzare pacchetti di installazione multilingue, generare aggiornamenti e racchiudere più file in un unico `.exe` per una distribuzione online semplice.*

Per creare un programma di installazione:

1. Fare clic su **Generate Installer** nella scheda "Distribute" di XLS Padlock. Si aprirà la finestra "Make Setup for your compiled workbook".



1. Compilare i campi "Destination path" (cartella di installazione predefinita), "Setup title" (titolo della finestra del programma di installazione) e "Your Application Name" (per i collegamenti, il programma di

disinstallazione, ecc.).

2. Premere **Generate Installer** per creare il progetto Paquet Builder. XLS Padlock avvierà quindi Paquet Builder, consentendo di personalizzare ulteriormente il progetto e di compilare il programma di installazione finale.

**Make installer for your compiled workbook** ✕

You can optionally distribute your compiled workbook .exe file in an **installer**: the installer can [create shortcuts](#) in the Windows Start menu folder, display a readme or license agreement, and offer an [uninstall option](#) to let your end users remove your application from their computer...

**Important:** XLS Padlock uses Paquet Builder to generate the installer; Paquet Builder must be installed on your computer. For further information about Paquet Builder, please click [here](#).

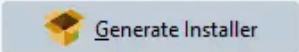
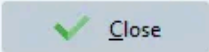
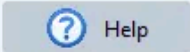
Please fill in the following settings and click **Generate Installer**. A **default project file** to be compiled with Paquet Builder will be created: simply open it in Paquet Builder, modify any settings if needed, and compile it.

Destination Path: c:\program files\

Installer Title:

Your Application Name (used for shortcuts and uninstall display name):

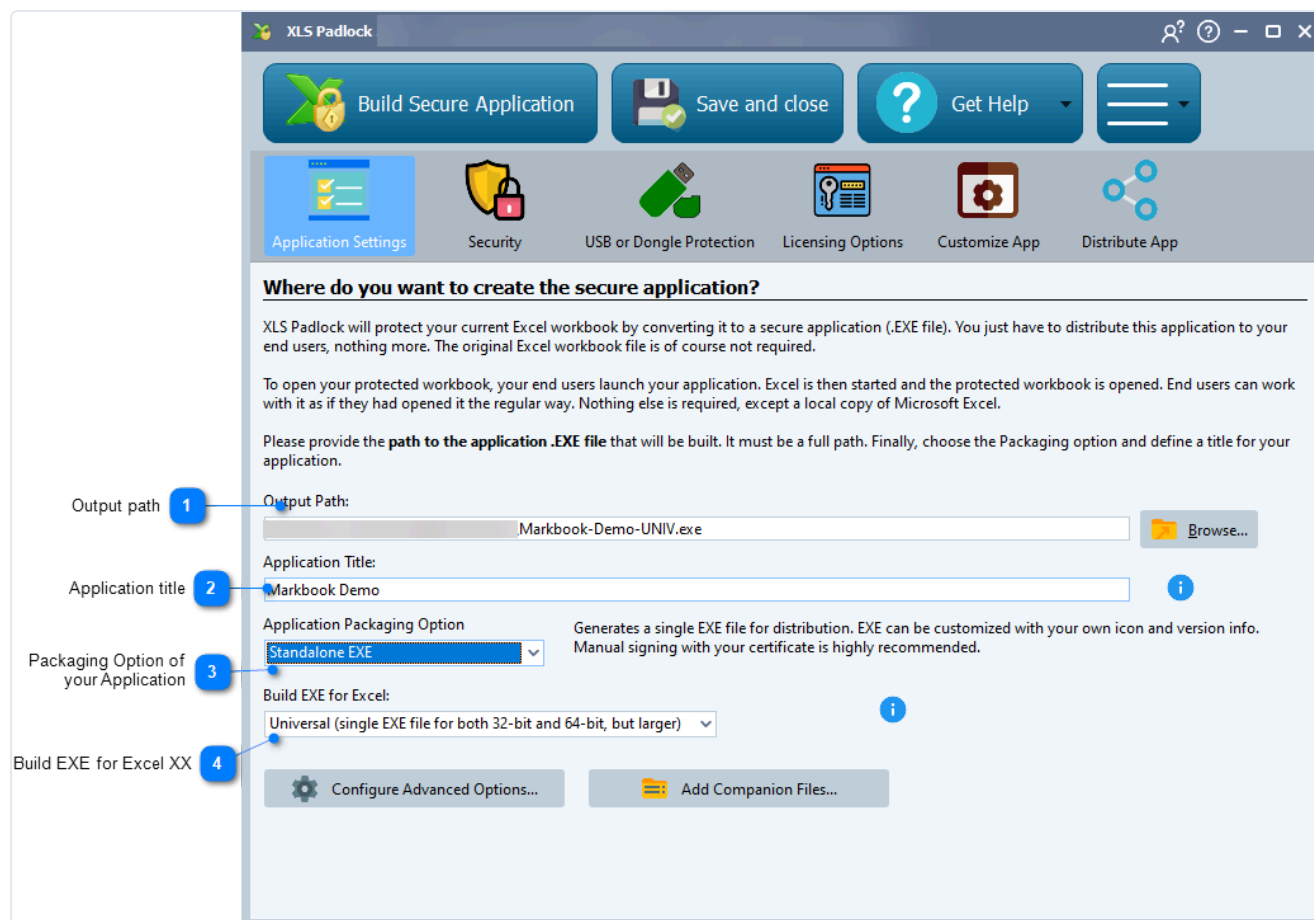
Open the generated project in Paquet Builder

👉 Consultare il [tutorial completo](#) su come creare un programma di installazione per la propria cartella di lavoro Excel.

# Compilare l'EXE in base all'architettura di Excel

Microsoft Excel è disponibile in due architetture: a 32 bit e a 64 bit. Poiché i 64 bit sono ormai lo standard per le versioni moderne di Office, la maggior parte degli utenti avrà bisogno di un'applicazione compatibile a 64 bit. Tuttavia, alcuni potrebbero ancora utilizzare la versione a 32 bit, in particolare su sistemi più datati. XLS Padlock consente di compilare per entrambe le piattaforme.



Questa opzione è disponibile solo per i [file eseguibili autonomi](#).

👉 Scegliere il tipo di file EXE che XLS Padlock genererà:

- **32-bit only**: Genera un file EXE compatibile solo con le versioni a 32 bit di Excel.
- **64-bit only**: Genera un file EXE compatibile solo con le versioni a 64 bit di Excel.
- **Universal**: Crea un unico file EXE che funziona sia con le versioni a 32 bit sia con quelle a 64 bit di Excel. Questo semplifica la distribuzione, ma il file EXE risultante avrà dimensioni all'incirca doppie.
- **32-bit and 64-bit**: Genera contemporaneamente due file EXE distinti. Per impedire che si sovrascrivano a vicenda, XLS Padlock aggiunge automaticamente i suffissi "32" e "64" ai nomi dei file (ad esempio, `MyApp32.exe` e `MyApp64.exe`).

### Incompatibilità di versione

Se un utente finale esegue un EXE con un'architettura di Office non corrispondente (ad esempio, un EXE a 64 bit su un'installazione di Office a 32 bit), riceverà un messaggio di errore che lo invita a contattare l'utente per ottenere la versione corretta.

### Firma del codice consigliata per la modalità Universal

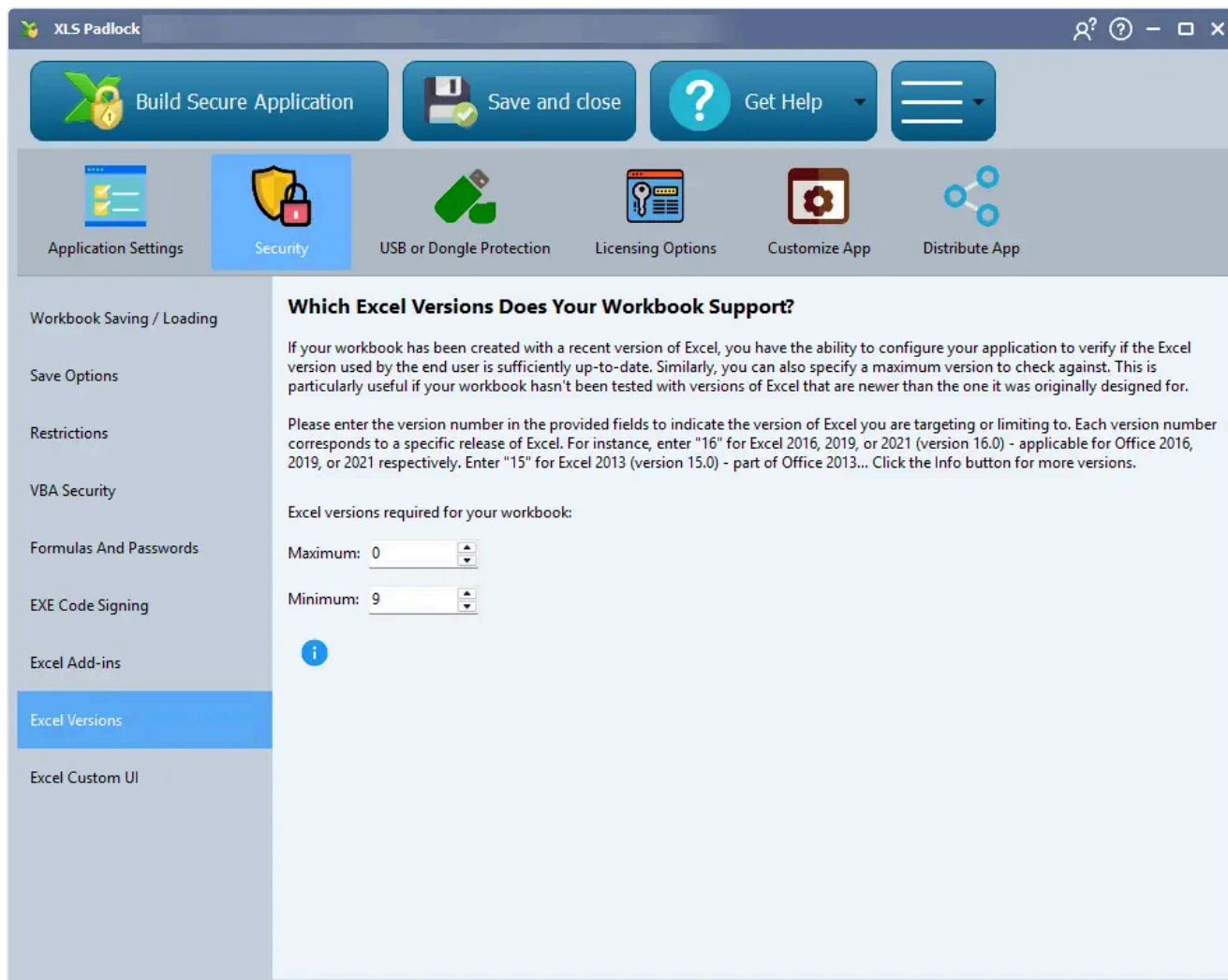
Se si utilizza la modalità Universal, si consiglia vivamente di firmare digitalmente l'applicazione. Questo perché, al momento dell'esecuzione, viene creato un file EXE temporaneo che a volte può essere segnalato dai software antivirus. Una firma digitale riduce notevolmente il rischio di falsi positivi.

### Info

Per impostazione predefinita, le [applicazioni distribuite in formato bundle](#) sono già universali.

# Versioni di Excel

Se la cartella di lavoro richiede funzionalità di una versione specifica di Excel, è possibile configurare l'applicazione affinché verifichi la versione di Excel dell'utente finale in fase di esecuzione.



È possibile impostare sia una versione minima richiesta sia una versione massima consentita. Questo è utile per garantire la compatibilità o per impedire l'uso su versioni di Excel più recenti e non testate.

Inserire il numero di versione corrispondente nei campi:

- **9:** Excel 2000 (versione 9.0)
- **10:** Excel 2002 (versione 10.0)
- **11:** Excel 2003 (versione 11.0)
- **12:** Excel 2007 (versione 12.0)
- **14:** Excel 2010 (versione 14.0)
- **15:** Excel 2013 (versione 15.0)
- **16:** Excel 2016, 2019, 2021 e 365 (versione 16.0)

# Aggiornamenti della cartella di lavoro

---

Dopo aver distribuito la sua applicazione basata su una cartella di lavoro Excel, prima o poi sarà necessario fornire aggiornamenti ai suoi utenti. XLS Padlock offre diversi strumenti per semplificare questo processo.

Un aspetto importante da considerare per gli aggiornamenti è il modo in cui la sua applicazione salva i dati dell'utente. È bene conoscere le [diverse modalità di salvataggio offerte da XLS Padlock](#) prima di generare l'applicazione.

Se la sua cartella di lavoro viene aggiornata di frequente e i suoi utenti devono modificare solo alcuni valori di cella, la modalità **Save defined cell values only (.XLSCE file)** è la scelta migliore. Questa modalità consente di aggiornare il file principale della cartella di lavoro senza compromettere i dati dell'utente, poiché XLS Padlock salva e ripristina [solo i valori di cella che ha designato](#).

Quando aggiorna la cartella di lavoro di origine, il modo più diretto per distribuire l'aggiornamento è ricompilarla e inviare il nuovo file EXE ai suoi clienti.

## Flusso di lavoro di aggiornamento consigliato

---

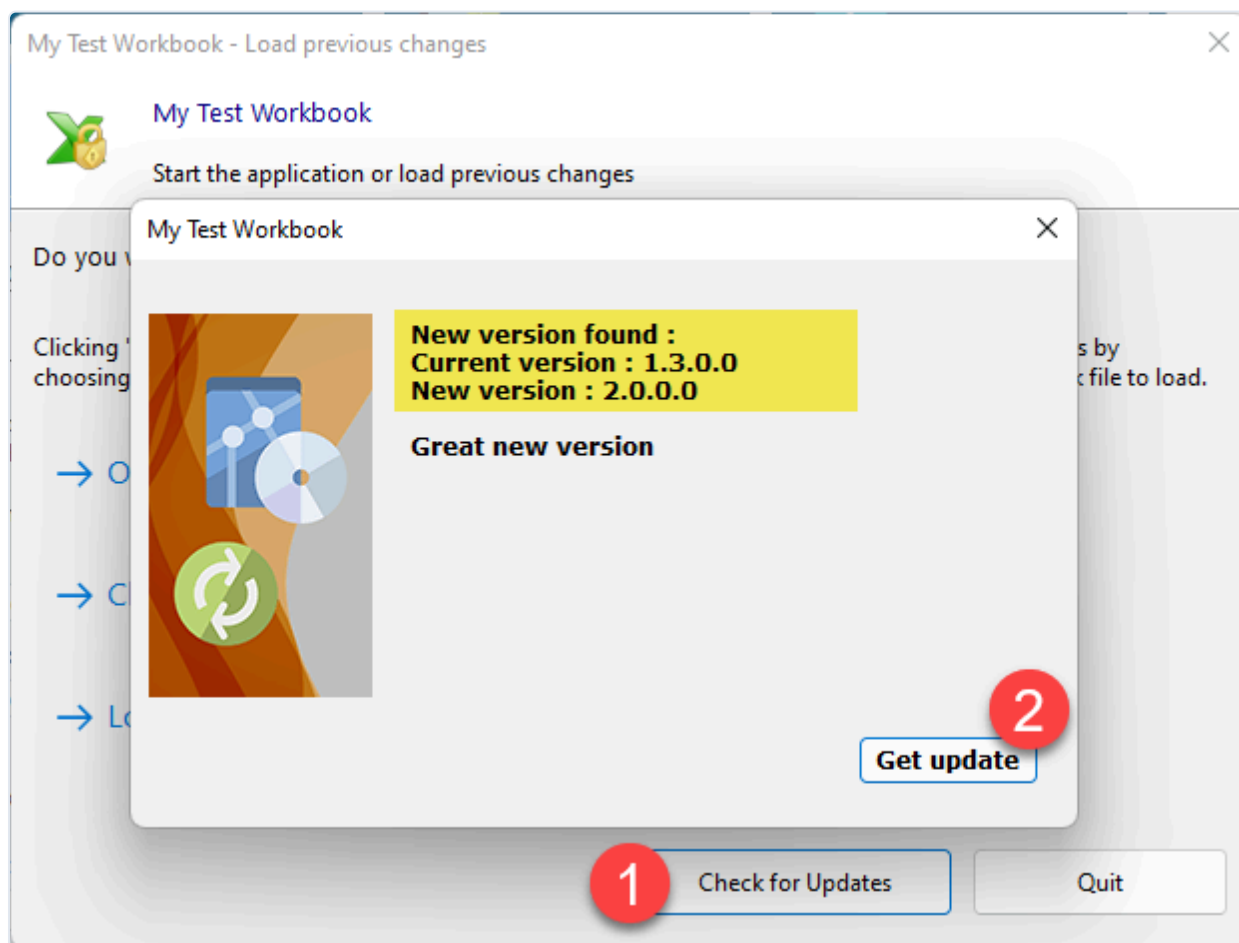
Il comportamento dal lato dell'utente dipende da ciò che ha modificato tra una versione e l'altra:

- **Solo il contenuto della cartella di lavoro** (formule, formattazione, fogli, logica VBA): mantenga l'**Application GUID** e la **Secret Key** esistenti. In modalità `.XLSCE`, gli utenti vedranno la sua nuova logica con i valori di cella salvati ripristinati. In modalità `.XLSC`, gli utenti continueranno a vedere l'ultima istantanea salvata, le sue modifiche appariranno solo se scelgono manualmente "Original Workbook" dalla schermata di benvenuto.
- **Vuole invalidare completamente i vecchi file di salvataggio degli utenti**: generi una nuova **Secret Key** nelle impostazioni [Activation and Licensing](#), ma lasci invariato l'**Application GUID** affinché le chiavi di attivazione esistenti rimangano valide. Al primo avvio da parte dell'utente, il suo vecchio file di salvataggio verrà rilevato come incompatibile e gli verrà proposta, tramite la schermata di benvenuto, la possibilità di caricare la cartella di lavoro originale incorporata.
- **Non cambi modalità di salvataggio tra una versione e l'altra** (`.XLSC` ↔ `.XLSCE`). I file di salvataggio esistenti generati con la modalità precedente verranno ignorati e qualsiasi dato che i suoi utenti non avevano ancora reinserito andrà perso. Scegli una modalità di salvataggio all'inizio del progetto e la mantenga per tutta la durata di vita dell'applicazione.

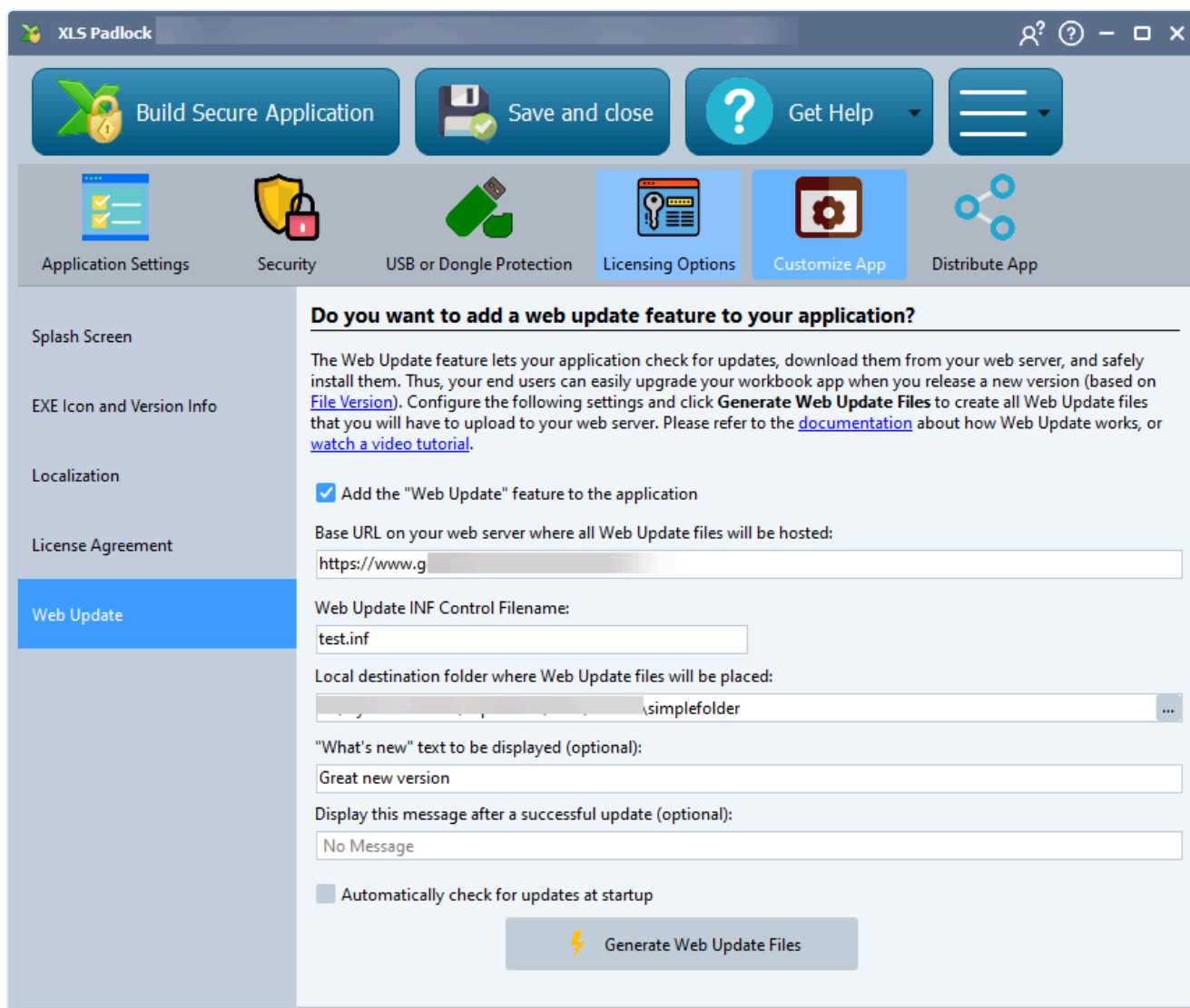
### SUGGERIMENTO

Il [pulsante Original Workbook nella schermata di benvenuto](#) funge da rete di sicurezza per l'utente quando un vecchio file di salvataggio non riesce a caricarsi. Anche se lo ha nascosto, XLS Padlock lo riattiverà automaticamente per il nuovo tentativo, in modo che gli utenti non rimangano mai bloccati.

👉 Tuttavia, scaricare e sostituire manualmente il file EXE può risultare laborioso per gli utenti finali. Per semplificare questa operazione, XLS Padlock offre una funzionalità di aggiornamento automatico via web. Può configurare la sua applicazione affinché verifichi online la disponibilità di aggiornamenti e avvisi gli utenti quando è disponibile una nuova versione:



Questa funzionalità può essere configurata rapidamente e direttamente in XLS Padlock:



👉 Per impostare il suo sistema di aggiornamento, avrà bisogno di un server web o di uno spazio di hosting in cui poter collocare i file per il download diretto. Segua poi le istruzioni della pagina [come configurare gli aggiornamenti automatici via web](#).

### Tutorial video

Guardi il nostro tutorial video su [come configurare gli aggiornamenti automatici via web per le sue cartelle di lavoro Excel](#).

# Aggiornamenti web automatici

XLS Padlock include una funzionalità di aggiornamento via web che può scaricare e installare automaticamente nuove versioni dell'applicazione. Questo consente agli utenti finali di aggiornare facilmente l'applicazione della cartella di lavoro ogni volta che si pubblica una nuova versione.

Non è richiesto alcun software di terze parti, ma **è necessario disporre di un server web o di uno spazio di hosting web** per ospitare i file di aggiornamento.

## Tutorial video

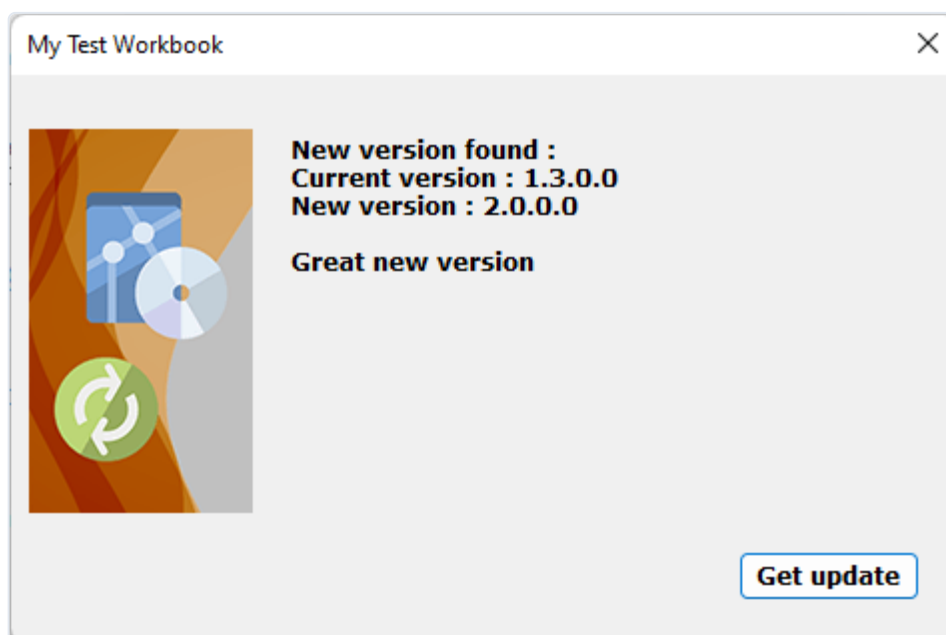
Guarda un tutorial video su come configurare gli aggiornamenti automatici via web per le cartelle di lavoro Excel.

## Come funziona l'aggiornamento via web

L'applicazione scarica prima un piccolo file di controllo ( `.inf` ) dal server. Questo file indica all'applicazione se è disponibile una nuova versione. In caso affermativo, l'applicazione scarica i file di aggiornamento necessari ( `.cab` ), si chiude, installa l'aggiornamento e si riavvia.

## Verifica degli aggiornamenti

È possibile configurare l'applicazione affinché **verifichi automaticamente la presenza di aggiornamenti all'avvio** oppure aggiungere un pulsante "**Check for Updates**" alla [schermata di benvenuto](#). Quando viene rilevata una nuova versione, gli utenti vengono guidati attraverso una procedura guidata di aggiornamento via web.



Gli utenti possono anche avviare un aggiornamento tramite l'[opzione da riga di comando](#) `-webupdate` .

## Come viene rilevata una nuova versione

Il numero di **File Version**, specificato nella pagina [EXE Version Info](#) di XLS Padlock, viene utilizzato per determinare se un aggiornamento è necessario. È necessario aumentare questo numero di versione ogni volta che si pubblica un aggiornamento.

## Configurazione

- **Base URL:** l'URL sul server web dove verranno ospitati i file di aggiornamento via web (ad esempio, `https://www.yourwebsite.com/myfolder` ). Si consiglia HTTPS.
- **Web Update INF Control Filename:** il nome del file di controllo che contiene le informazioni sull'aggiornamento.
- **Local Destination Folder:** una cartella locale sul computer dove XLS Padlock genererà i file di aggiornamento. Sarà quindi necessario caricare l'intero contenuto di questa cartella sul server web.

### Avvertenza

La cartella di destinazione deve essere vuota. XLS Padlock chiederà di eliminarne il contenuto se non lo è.

- **Testo "What's new" (facoltativo):** testo da visualizzare nella procedura guidata di aggiornamento, che informa gli utenti sulle modifiche presenti nella nuova versione.

## Generazione dei file di aggiornamento via web

Facendo clic su **Generate Web Update Files** in XLS Padlock verranno creati i file necessari (un file di controllo `.inf` e un file compresso `.cab` ) nella cartella locale specificata. Sarà quindi possibile caricare questi file sul server web.

### Privilegi di amministratore

Se l'applicazione è installata in una cartella di Windows con restrizioni come `Program Files` , saranno necessari privilegi di amministratore per installare l'aggiornamento, il che attiverà un avviso UAC.

## Risoluzione dei problemi

Per risolvere i problemi della funzionalità di aggiornamento via web, attivare l'opzione "**Enable WebUpdate Log**" nelle [Opzioni avanzate](#). Questo creerà un file `WUPDATE.LOG` nella cartella Documenti dell'utente, contenente informazioni dettagliate sul processo di aggiornamento.

# Migrare i dati utente tra gli aggiornamenti

Quando si aggiorna la cartella di lavoro protetta e si distribuisce un nuovo EXE, si possono incontrare difficoltà nel garantire il corretto trasferimento dei dati degli utenti, soprattutto se si utilizza la [modalità Full Save](#). Questa modalità crea un'**istantanea completa della cartella di lavoro** al momento del salvataggio. Di conseguenza, quando un utente apre il proprio vecchio file di salvataggio ( `.xlsc` ) con la nuova applicazione, vedrà i propri vecchi dati nella vecchia struttura della cartella di lavoro e non beneficerà dei vostri ultimi aggiornamenti (come nuove funzionalità o correzioni di bug nel vostro codice VBA).

Questo argomento spiega come utilizzare un sistema di esportazione/importazione basato su VBA per aiutare gli utenti a migrare i propri dati da una vecchia versione dell'applicazione a una nuova.

## Avvertenza

La soluzione qui descritta dipende dalla [modalità di salvataggio selezionata per il progetto XLS Padlock](#). Se si utilizza la modalità di salvataggio **Cell Values**, i dati utente vengono caricati automaticamente nella struttura della cartella di lavoro aggiornata e questo processo manuale non è necessario.

## Alternativa: modalità di salvataggio Cell Values

Come indicato sopra, la [modalità di salvataggio Cell Values](#) è il modo più semplice per gestire gli aggiornamenti. Tuttavia, questa modalità è progettata per salvare solo i valori di celle specifiche. Se occorre trasferire altri tipi di dati, come lo stato di variabili o oggetti VBA, sarà necessario utilizzare una soluzione VBA personalizzata come quella descritta in questo argomento.

## ## Esempio di codice

Questo processo comporta due passaggi principali: esportare i dati dalla vecchia versione e importarli nella nuova.

### 1. Esportare i dati utente in un file Excel

La macro seguente genera un normale file Excel `.xlsx` contenente i dati dell'utente. È necessario includere questa macro nella vecchia applicazione, associata a un pulsante, in modo che gli utenti possano esportare i propri dati prima di eseguire l'aggiornamento.

#### NOTA

Sarà necessario sostituire i nomi dei fogli e gli intervalli di celle nella macro seguente affinché corrispondano alla struttura della vostra cartella di lavoro.

```
Sub GenerateData()
 Dim savePath As String
 'New workbook with 3 sheets
 Workbooks.Add xlWBATWorksheet
 ActiveSheet.Name = "SheetA"
 Sheets.Add(After:=Sheets(1)).Name = "SheetB"
 Sheets.Add(After:=Sheets(2)).Name = "SheetC"
 ActiveWorkbook.Sheets("SheetA").Range("A1:C3").Value = ThisWorkbook.Sheets("SheetA").Range(
ActiveWorkbook.Sheets("SheetB").Range("B3").Value = ThisWorkbook.Sheets("SheetB").Range("B3
ActiveWorkbook.Sheets("SheetC").Range("B1:C3").Value = ThisWorkbook.Sheets("SheetC").Range(
savePath = Application.GetSaveAsFilename("", "Excel workbook (*.xlsx),*.xlsx", 1, "Export U
If savePath <> "False" Then ActiveWorkbook.SaveAs savePath, FileFormat:=51
 ActiveWorkbook.Close False
End Sub
```

## 2. Caricare i dati nel nuovo file EXE.

L'utente deve aprire il nuovo EXE ed eseguire la 3<sup>a</sup> macro seguente per caricare i dati (associa la macro a un pulsante). Una volta che l'utente esegue la macro, gli verrà chiesto di selezionare il file (1<sup>a</sup> macro) e i dati verranno copiati (2<sup>a</sup> macro). La terza macro esegue entrambe le macro, ed è questa la macro che deve essere associata al pulsante. Anche in questo caso, sarà necessario modificare le celle e i nomi delle schede nella seconda macro, ed è possibile modificare anche il titolo nella prima macro:

```
***** 1st macro:
Sub Open_Workbook_Dialog()
Dim my_FileName As Variant
my_FileName = Application.GetOpenFilename(_
FileFilter:="Excel Files,.xl;.xm", _
FilterIndex:=3, _
Title:="Select the old version of your file, where you will pull the data from", _
MultiSelect:=False)
If my_FileName <> False Then
Workbooks.Open Filename:=my_FileName
End If
End Sub

***** 2nd macro:
Sub TransferData()
If Workbooks.Count > 1 Then
Workbooks(1).Sheets("SheetA").Range("A1:C3").Value = Workbooks(2).Sheets("SheetA").Range("A1:C3").Value
Workbooks(1).Sheets("SheetB").Range("B3").Value = Workbooks(2).Sheets("SheetB").Range("B3").Value
Workbooks(1).Sheets("SheetC").Range("B1:C3").Value = Workbooks(2).Sheets("SheetC").Range("B1:C3").Value
Workbooks(2).Close savechanges:=False
Else
MsgBox "The data hasn't been transferred.", vbExclamation, "Error"
End If
End Sub

***** 3rd macro:
Sub TheTransfer()
Call Open_Workbook_Dialog
Call TransferData
End Sub
```

**NOTA**

Per salvare cartelle di lavoro normali, è possibile consultare anche: [Caricamento/salvataggio di cartelle di lavoro tramite l'helper VBA SetOption](#)

👉 Vedere anche:

- [Modalità di salvataggio selezionata per il progetto XLS Padlock](#)

# Salvare e ripristinare le impostazioni tramite modelli

---

XLS Padlock consente di salvare le impostazioni correnti del progetto in un file modello. Si tratta di una potente funzionalità che può far risparmiare una notevole quantità di tempo, soprattutto se si creano frequentemente nuove applicazioni di cartelle di lavoro con configurazioni simili.

## Salvare le impostazioni del progetto in un modello

---

Una volta configurate tutte le opzioni desiderate per il progetto (come le impostazioni di sicurezza, la personalizzazione e le opzioni di output), è possibile salvare questa configurazione come modello.

Fare clic sul pulsante



App Menu e quindi sul menu **Save Current Settings As Template**. Verrà richiesto di scegliere una posizione e un nome per il file modello (che avrà estensione `.XPLP`).

### Avvertenza

Non scegliere il file di progetto XLS Padlock esistente.

## Ripristinare le impostazioni da un modello

---

Quando si avvia un nuovo progetto o si desidera applicare una configurazione standard a un progetto esistente, è possibile ripristinare facilmente le impostazioni da un modello salvato in precedenza.

Fare clic sul pulsante



App Menu, quindi sul menu **Restore Project from Template**, individuare il file modello `.XPLP` e selezionarlo. Tutte le impostazioni memorizzate in quel modello verranno applicate istantaneamente al progetto corrente.

### NOTA

Il percorso di output dell'applicazione non viene memorizzato nei modelli.

# Errore del registro o EREGISTRYEXCEPTION

---

Questo errore indica spesso un conflitto con un software di sicurezza o un programma antivirus. Per risolverlo, attivare l'opzione « **Use another registry key for storing activation data** » (utilizzare un'altra chiave di registro per memorizzare i dati di attivazione) nelle [Advanced Options](#).

# Violazione di accesso all'indirizzo

---

Questo errore può verificarsi quando si utilizza la funzionalità « Lock VBA Project (simple VBA protection) ». Per risolverlo, rimuovere la password dal progetto VBA prima di compilare la cartella di lavoro con XLS Padlock. La password non è necessaria, poiché XLS Padlock bloccherà il progetto automaticamente senza richiederne una.

In alternativa, è possibile provare a utilizzare una password più breve oppure passare all'opzione [Prevent access to VBA editor](#).

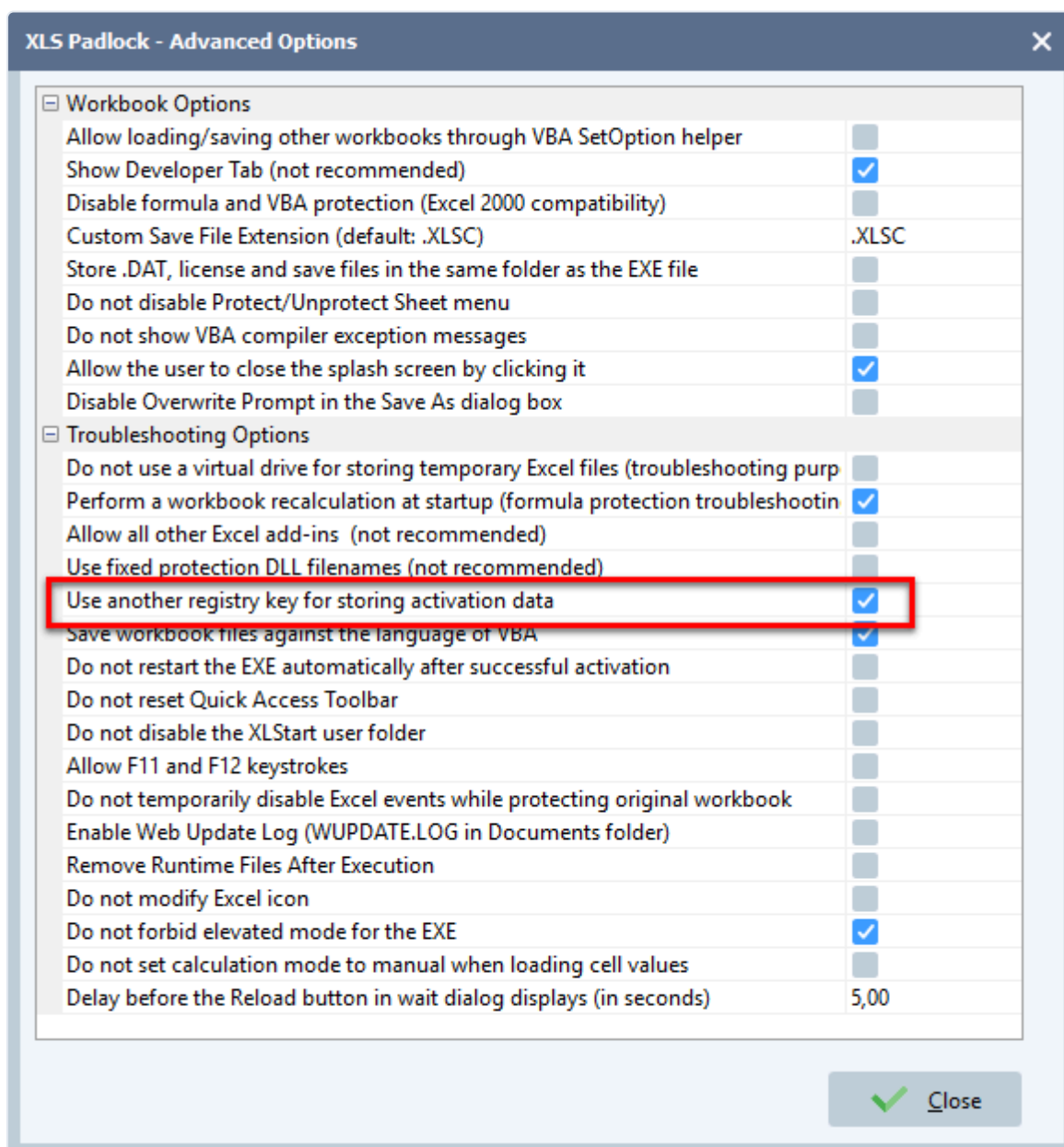
# Errore "Failed to set data for 'Data'"

Se l'utente finale riceve il seguente messaggio di errore durante l'immissione di una chiave di attivazione, di solito indica un conflitto con un software di sicurezza sul suo sistema.

## Error setting data in registry, Failed to set data for 'Data'

Questo errore si verifica perché un altro programma, di solito un antivirus o una suite di sicurezza, impedisce all'applicazione di scrivere i propri dati di attivazione nel Registro di sistema di Windows.

Per risolvere il problema, è possibile configurare il proprio progetto XLS Padlock in modo che utilizzi una chiave di registro alternativa. Abilitare l'opzione avanzata **Use another registry key for storing activation data** (utilizzare un'altra chiave di registro per memorizzare i dati di attivazione), come illustrato di seguito, quindi ricompilare l'applicazione.



# Perché l'EXE è così grande

---

Il file `.exe` compilato è notevolmente più grande della cartella di lavoro originale perché è un'applicazione autonoma che racchiude diversi componenti essenziali:

- Un loader sicuro compatibile con varie versioni di Excel e di Windows.
- Un motore di esecuzione completo con supporto Unicode.
- Il compilatore e l'interprete VBA proprietari.
- Diversi livelli di protezione anti-pirateria e di protezione del codice.

Per ridurre la dimensione del file EXE:

- **Use UPX Compression** (utilizzare la compressione UPX): UPX è un compressore di EXE gratuito e open source, disponibile su [upx.github.io](https://upx.github.io). La compressione dell'EXE può ridurre notevolmente la dimensione. Si consiglia tuttavia vivamente di firmare digitalmente qualsiasi file EXE compresso, poiché in caso contrario alcuni programmi antivirus potrebbero generare falsi positivi.
- **Disable Formula Protection** (disattivare la protezione delle formule): se l'applicazione non si basa su formule sensibili, è possibile disattivare la protezione delle formule per ridurre leggermente la dimensione finale del file EXE.

# Ripristinare il file XLS dall'EXE

---

No, non è possibile ripristinare il file Excel originale da un .EXE compilato se la funzione "Save As" è stata disattivata.

Se è necessario poter recuperare la cartella di lavoro originale, occorre attivare la funzione "Save As" durante la compilazione. Questo consente di utilizzare la funzionalità di [decifrazione dei file salvati](#).

# Link di assistenza

---

Per qualsiasi domanda o problema con XLS Padlock, è possibile contattarci attraverso i seguenti canali:

- **Forum utenti:** <https://www.gdgsoft.info> (Scegliere la categoria "XLS Padlock")
- **Supporto via e-mail:** [info@xlspadlock.com](mailto:info@xlspadlock.com)
- **Per il supporto del dongle Enky SL/LC:** [info@hs-securityware.com](mailto:info@hs-securityware.com)

Quando si richiede assistenza, si prega di indicare la propria versione di Excel (ad esempio, Excel 365, Excel 2019) e l'architettura (32 bit o 64 bit) per aiutarci a risolvere il problema più rapidamente.

---

**Seguici su X:** <https://x.com/gdgsoft>

**I nostri altri prodotti sono disponibili su:** <https://www.gdgsoft.com>