

XLS PADLOCK

Guide de l'utilisateur XLS Padlock

Protection, licences et distribution de classeurs Excel

Version 2026.0

G.D.G. Software

www.xlspadlock.com

Copyright © G.D.G. Software 2018-2026. Tous droits réservés.

Table des matières

01	À propos de ce guide
02	Introduction à XLS Padlock
03	Télécharger et installer
04	Paramètres de l'application
05	XLS Padlock Manager
06	Comment protéger un classeur Excel
07	Sécurité
08	Options Cloud et IA
09	Améliorer la protection
10	Contrôle d'accès au classeur
11	Restrictions du classeur Excel
12	Protection par mot de passe des feuilles
13	Combiner la protection Excel et XLS Padlock
14	Protection des formules : Excel ou XLS Padlock
15	Protéger les formules avec XLS Padlock
16	Marquer les cellules à protéger
17	Méthode de protection des formules

18 Désactiver la protection des formules

19 À propos du VBA Compiler intégré

20 Véritable protection du code VBA

21 Référence de la syntaxe

22 Recettes de l'API VBA

23 Écrire et compiler du VBA sécurisé

24 Invoquer le VBA compilé à l'exécution

25 Accéder aux objets Excel

26 Passer des tableaux

27 Passer davantage de paramètres

28 Gérer les erreurs

29 Erreur OLE 800A03EC

30 Protection du code VBA

31 Masquer et verrouiller le code VBA

32 Verrouiller le projet VBA

33 Interdire l'accès au VBE

34 Guide d'activation et de licence

35 Clés d'activation

36 Configurer les clés d'activation

- [37 Clés d'activation verrouillées au matériel](#)

- [38 Activation en ligne](#)

- [39 Validation en ligne](#)

- [40 Éditeur du formulaire d'enregistrement](#)

- [41 Afficher le contrat de licence](#)

- [42 Générateur de clés \(portable et serveur\)](#)

- [43 SDK du générateur de clés](#)

- [44 Générateur de clés autonome](#)

- [45 Restrictions sur les clés](#)

- [46 Protection par USB ou dongle](#)

- [47 Désactivation](#)

- [48 Créer des classeurs d'essai](#)

- [49 Fermer après un délai donné](#)

- [50 Vérifier l'état d'essai](#)

- [51 Jours d'essai restants](#)

- [52 Options d'enregistrement du classeur](#)

- [53 Enregistrement et chargement](#)

- [54 Mode d'enregistrement : complet ou valeurs de cellules](#)

- [55 Définir les cellules à enregistrer et restaurer](#)

- 56 Restaurer et enregistrer via VBA

- 57 Accéder au classeur sécurisé et aux fichiers compagnons

- 58 Ajouter des fichiers compagnons

- 59 Dossier de stockage des enregistrements

- 60 Stocker les modifications dans l'EXE

- 61 Restreindre le chargement et l'enregistrement

- 62 Ouvrir et déchiffrer les enregistrements

- 63 Verrouiller les enregistrements à une machine

- 64 Références externes et hyperliens

- 65 Obtenir le chemin à côté du classeur compilé

- 66 Personnaliser l'application

- 67 Empaquetage de l'application

- 68 Écran de démarrage

- 69 Changer l'icône de l'EXE

- 70 Fenêtre Excel au démarrage

- 71 Exécuter en application VBA uniquement

- 72 Informations de version de l'EXE

- 73 Commutateurs de ligne de commande

- 74 Masquer la boîte de dialogue de chargement

- [75 Localisation et traduction](#)

- [76 Options avancées](#)

- [77 Ignorer les erreurs de traitement](#)

- [78 Désactiver les informations de débogage](#)

- [79 Interface Excel personnalisée](#)

- [80 Compléments Excel](#)

- [81 Distribuer l'application](#)

- [82 Distribuer un classeur protégé](#)

- [83 Signer numériquement l'EXE](#)

- [84 Créer un programme d'installation](#)

- [85 Compiler l'EXE selon l'architecture d'Excel](#)

- [86 Versions d'Excel](#)

- [87 Mises à jour du classeur](#)

- [88 Mises à jour web automatiques](#)

- [89 Migrer les données utilisateur entre les mises à jour](#)

- [90 Enregistrer et restaurer les réglages via des modèles](#)

- [91 Erreur de registre ou EREGISTRYEXCEPTION](#)

- [92 Violation d'accès à l'adresse](#)

- [93 Erreur « Failed to set data for 'Data' »](#)

94 Pourquoi l'EXE est-il si volumineux

95 Restaurer le fichier XLS depuis l'EXE

96 Liens d'assistance

À propos de ce guide

Copyright © G.D.G. Software 2013-2026

Tous droits réservés. Aucune partie de cet ouvrage ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit, graphique, électronique ou mécanique, y compris la photocopie, l'enregistrement, l'archivage sur bande ou les systèmes de stockage et de récupération de l'information, sans l'autorisation écrite de l'éditeur.

Les clients disposant d'une licence XLS Padlock sont autorisés à imprimer ce guide pour un usage privé ou éducatif.

Microsoft Excel® et Office® sont des marques déposées de Microsoft Corporation.

Tous les produits mentionnés dans ce document peuvent être des marques commerciales ou des marques déposées de leurs propriétaires respectifs. L'éditeur et l'auteur ne revendiquent aucun droit sur ces marques.

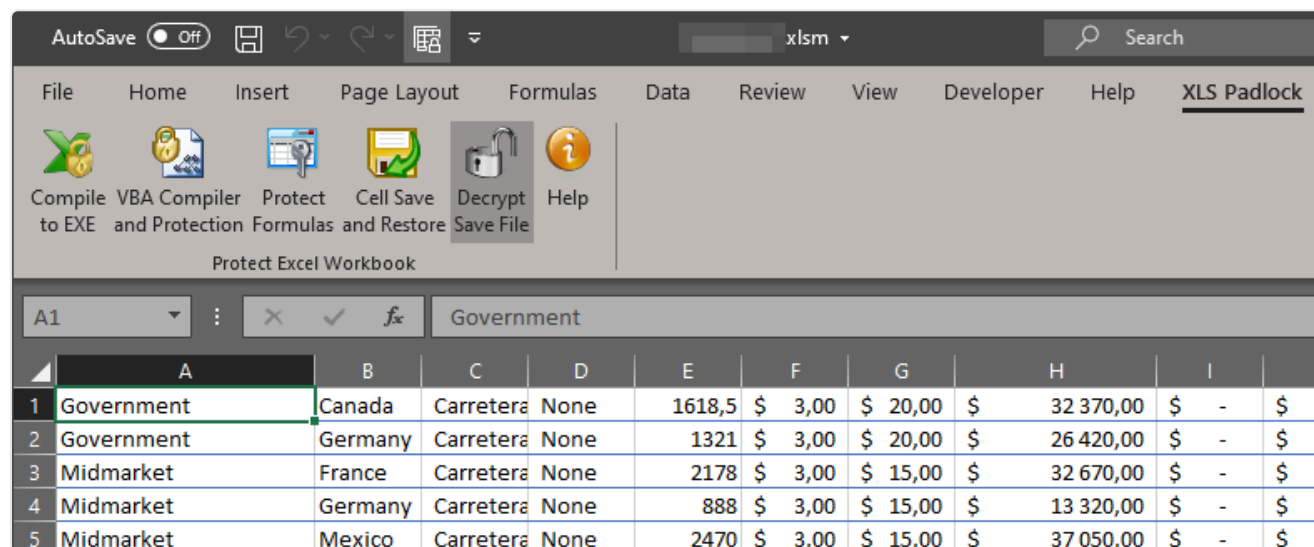
Bien que toutes les précautions aient été prises lors de la préparation de ce document, l'éditeur et l'auteur déclinent toute responsabilité quant aux erreurs ou omissions, ou aux dommages résultant de l'utilisation des informations qu'il contient ou de l'utilisation des programmes et du code source qui l'accompagnent. En aucun cas l'éditeur ou l'auteur ne pourront être tenus responsables d'une quelconque perte de bénéfices ou de tout autre dommage commercial causé ou prétendument causé directement ou indirectement par ce document.

Première édition : janvier 2013. Mise à jour : juin 2026.

Version 2026.0, [Lire le journal des modifications](#)

Introduction à XLS Padlock

XLS Padlock est **une puissante solution de protection et de gestion de licences pour Microsoft Excel**, conçue pour protéger vos classeurs contre la copie. Il fonctionne comme un compilateur, ce qui vous permet de convertir vos classeurs Excel en applications autonomes et sécurisées.



Cela vous permet de distribuer vos fichiers Excel en toute sécurité, de contrôler l'accès des utilisateurs et d'empêcher la copie des feuilles de calcul. Des fonctionnalités clés d'Excel peuvent être restreintes, notamment :

- Le menu contextuel des feuilles de calcul (clic droit)
- La fonctionnalité copier/coller
- La barre de formule
- L'enregistrement et l'impression
- L'accès au projet VBA

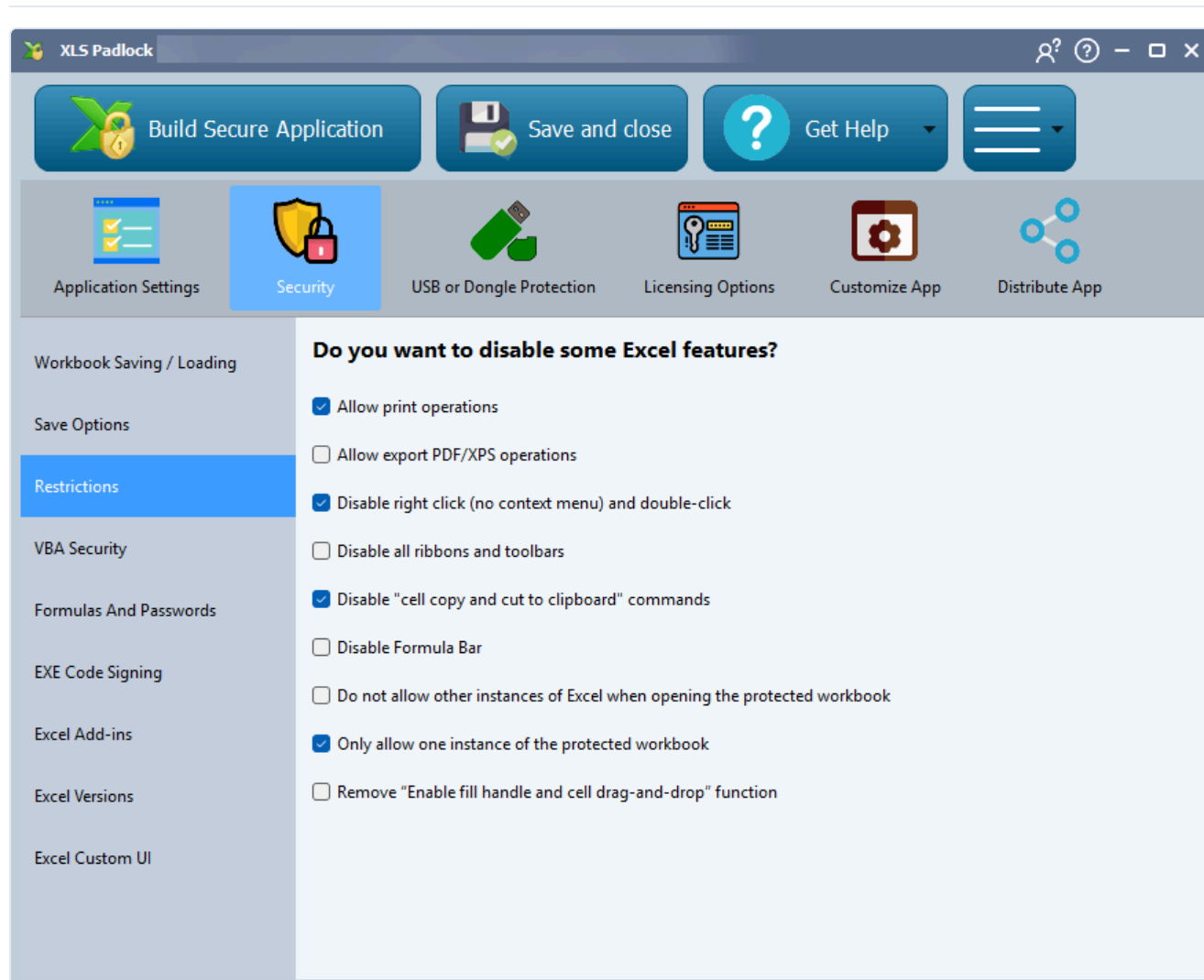
XLS Padlock offre plusieurs moyens de sécuriser vos fichiers, comme **les verrouiller sur une clé USB spécifique**, utiliser une **puissante protection par dongle**, ou émettre des **Activation Keys verrouillées sur le matériel**. Vous bénéficiez également d'un contrôle granulaire sur la protection des formules, ce qui permet aux utilisateurs d'utiliser vos formules sans pouvoir les afficher ni les copier.

Protection avancée du code VBA

Grâce à son VBA Compiler intégré, XLS Padlock vous permet de **compiler du code VBA sensible en byte-code sécurisé, le rendant inaccessible aux utilisateurs finaux**. Vos macros VBA sont véritablement protégées, car le code source d'origine est supprimé.

Le compilateur n'est **pas un simple obfuscateur** ; il transforme le code VBA en code binaire et le stocke de manière sécurisée dans l'EXE de l'application. Pour compléter la protection, vous pouvez verrouiller votre projet VBA, **rendant inutiles les outils de crackage de mot de passe**. [En savoir plus sur la protection du code VBA.](#)

Options de sécurité complètes



Les applications peuvent être configurées pour **expirer après un nombre de jours défini**, un nombre d'utilisations précis, ou à une date fixe. Un écran de rappel personnalisable peut être affiché pour les versions d'essai. **L'activation en ligne** automatise le processus de gestion de licences, vous donnant un **contrôle à distance** sur les personnes autorisées à accéder à vos classeurs.

Les applications de classeur protégé sont autonomes et ne nécessitent que Microsoft Excel pour fonctionner. Toutes les fonctionnalités d'Excel sont entièrement prises en charge.

👉 Vous pouvez également personnaliser et marquer vos applications :

- Ajouter une icône personnalisée et des informations de copyright.
- Traduire tout le texte visible par l'utilisateur dans n'importe quelle langue.
- Afficher un contrat de licence (EULA) ou un écran de démarrage au lancement.
- Signer numériquement vos classeurs compilés avec Authenticode.
- Créer un programme d'installation professionnel pour la distribution.
- Avertir les utilisateurs des nouvelles versions grâce à la fonctionnalité intégrée de mise à jour web.

Compatible avec Office 365 et Excel 2024, 2021, 2019, 2016, 2013, 2010, 2007 (SP3), 2002, versions 32 bits et 64 bits.

Avertissement

Une version 64 bits de Windows est requise pour créer des applications 64 bits.

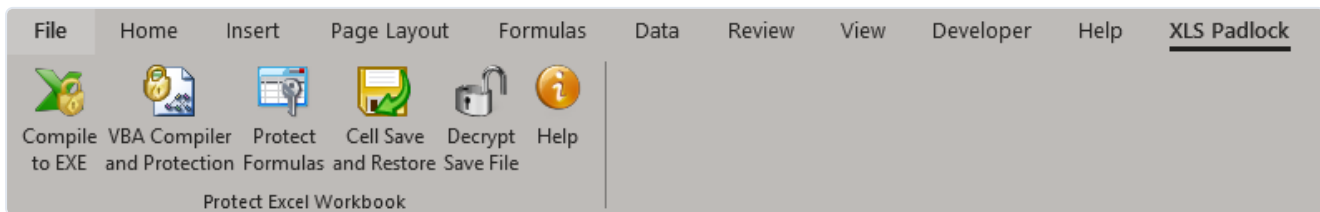
👉 À découvrir maintenant :

- [Télécharger et installer XLS Padlock](#)
- [Comment protéger un classeur Excel](#)
- [Site web de XLS Padlock](#)
- [Nouveautés](#)

Télécharger et installer

Sur notre site web à l'adresse <https://www.xlspadlock.com/download>, vous devez sélectionner le programme d'installation correspondant à votre version de Microsoft Excel (32 bits ou 64 bits). Le programme d'installation vérifiera que vous installez la version correcte.

Après une installation réussie, **XLS Padlock est intégré directement dans Excel**, apparaissant sous forme d'un nouvel onglet dans le ruban. Pour utiliser XLS Padlock, vous devez d'abord ouvrir votre classeur Excel.



Deux raccourcis sont également créés sur votre bureau : l'un pour ce guide de l'utilisateur et l'autre pour le XLS Padlock Manager.

- [XLS Padlock Manager](#)
- [Comment protéger un classeur Excel](#)

Paramètres de l'application

La page Application Settings contient les paramètres obligatoires de votre projet.

Paramètres principaux

Output Path

👉 Définit l'emplacement où le fichier EXE final sera créé.

XLS Padlock compile votre classeur en un seul fichier exécutable (EXE). Dans le champ "Output Path", vous devez spécifier le chemin complet et le nom de fichier où vous souhaitez que votre application soit créée.

CONSEIL

N'oubliez pas d'inclure l'extension `.exe` dans le nom de fichier.

Les chemins relatifs sont également acceptés ; ils seront relatifs au dossier qui contient votre fichier de classeur source. Par défaut, XLS Padlock propose un chemin dans le même dossier que votre classeur source, mais vous pouvez le remplacer par n'importe quel emplacement.

Application Title

👉 Définit le titre de la fenêtre principale de votre application.

Vous pouvez définir un titre personnalisé pour votre application. Ce titre est affiché dans la barre de titre de la fenêtre, remplaçant le texte par défaut "Microsoft Excel".

Si vous souhaitez afficher dynamiquement le nom de fichier d'un fichier de sauvegarde chargé, vous pouvez utiliser les espaces réservés suivants dans le titre :

- `%SAVEFILENAME%` : Remplacé par le nom de fichier uniquement (par exemple, `MyData.xlsx`).
- `%SAVEFULLNAME%` : Remplacé par le chemin complet vers le fichier de sauvegarde (par exemple, `C:\Users\Me\Documents\MyData.xlsx`).

Par exemple, un titre défini sur `My Application, %SAVEFILENAME%` s'afficherait sous la forme "My Application, MyData.xlsx" lorsque l'utilisateur a ce fichier ouvert.

REMARQUE

Si le classeur d'origine, non enregistré, est chargé, les espaces réservés seront vides.

Application Packaging Option

Voir [Packaging Option](#) : Choisissez entre un EXE autonome ou un bundle pour la distribution.

Build EXE for Excel

Voir [Build EXE for Excel](#) : Indiquez s'il faut compiler pour une compatibilité 32 bits, 64 bits ou universelle.

Configuration supplémentaire

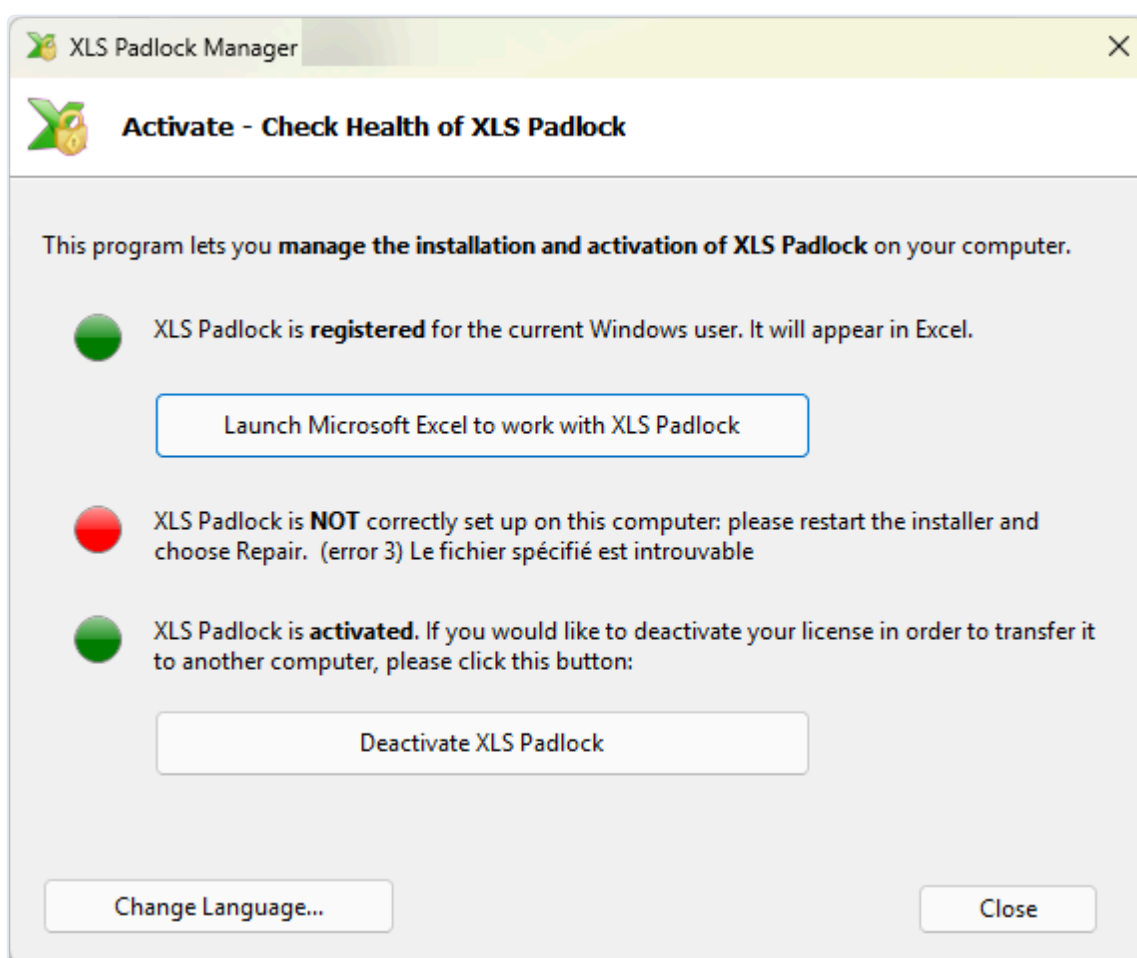
- [Configurer les options avancées](#)
- [Ajouter des fichiers compagnons](#)

XLS Padlock Manager

XLS Padlock Manager est une application autonome qui vous permet de **vérifier si XLS Padlock est correctement configuré et enregistré pour l'utilisateur Windows courant**.

XLS Padlock **doit être enregistré pour chaque utilisateur Windows qui souhaite l'utiliser**. Lorsque vous installez XLS Padlock sur un ordinateur, il n'est généralement enregistré que pour le compte administrateur qui a effectué l'installation. Les autres comptes utilisateur doivent utiliser le XLS Padlock Manager pour enregistrer le logiciel.

- Pour enregistrer XLS Padlock pour votre compte utilisateur Windows, lancez « XLS Padlock for Excel, Manager » depuis votre bureau Windows. La fenêtre suivante apparaîtra :



- Si tous les indicateurs sont verts, votre installation est correcte.
- Si le premier indicateur est rouge, cliquez sur « **Enable XLS Padlock for the current Windows user** ». XLS Padlock sera alors enregistré, et vous pourrez commencer à l'utiliser. Cette action ne nécessite pas de droits d'administrateur.



REMARQUE

Si vous disposez d'une licence XLS Padlock, vous pouvez également utiliser le manager pour l'activer.

👉 [Comment protéger un classeur Excel](#)

Comment protéger un classeur Excel

Tutoriel vidéo disponible

Découvrez ces tutoriels vidéo sur XLS Padlock réalisés par un expert Excel :

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

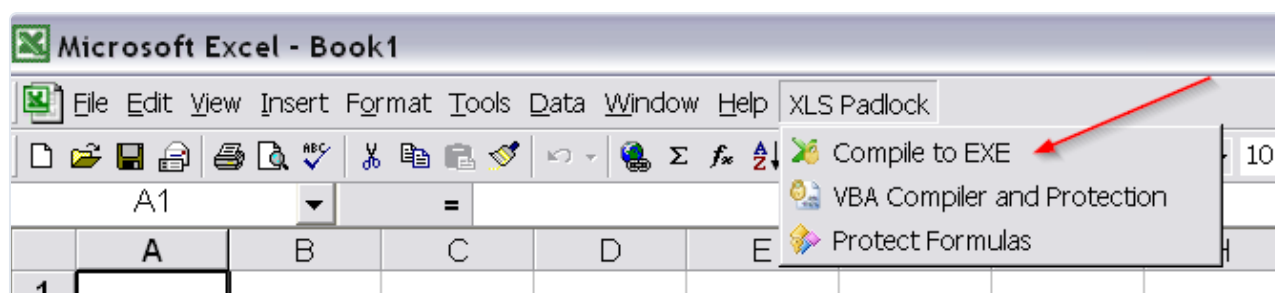
1\ Ouvrez et enregistrez votre classeur

Commencez par ouvrir dans Excel le classeur que vous souhaitez protéger. Assurez-vous que toutes les modifications sont enregistrées avant de continuer.

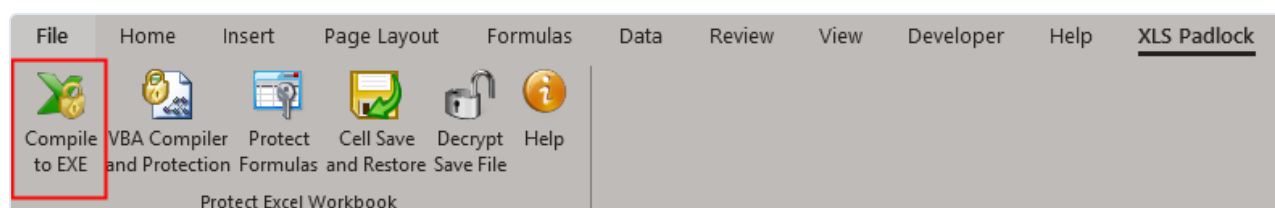
2. Configurez les options de protection

Pour commencer, ouvrez l'interface de XLS Padlock depuis le ruban Excel :

- **Avant Excel 2007** : utilisez le menu « XLS Padlock ».



- **Excel 2007 et versions ultérieures** : cliquez sur « Secure Compile to Exe » dans l'onglet « XLS Padlock » du ruban.



REMARQUE

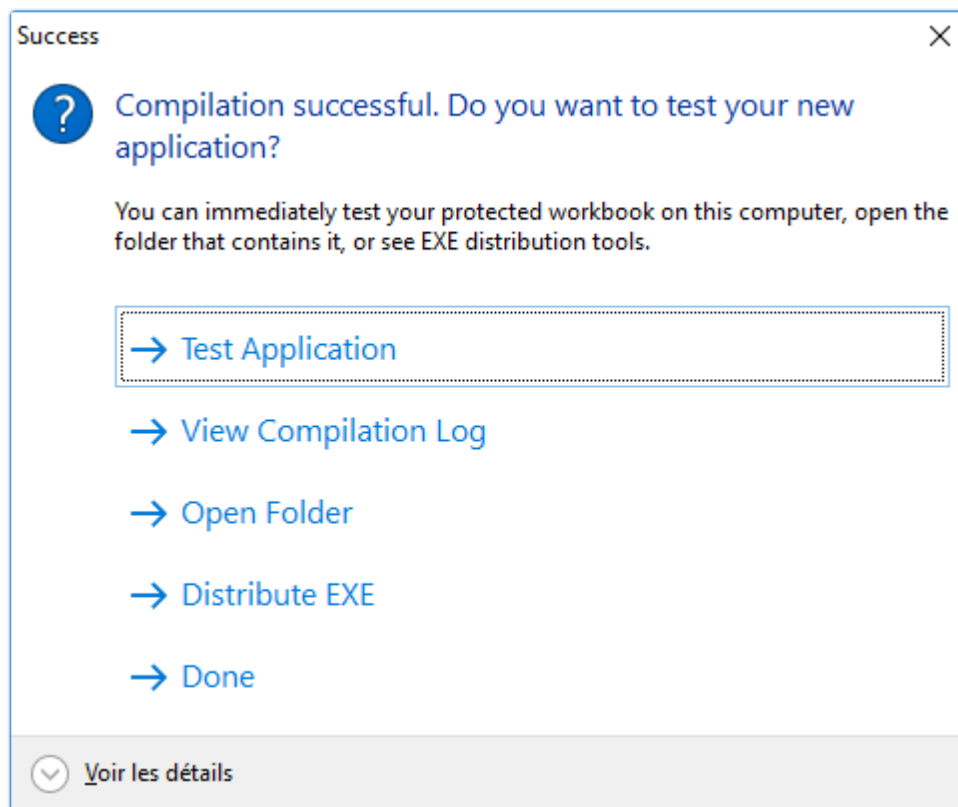
Enregistrez toujours votre classeur Excel avant d'ouvrir l'interface de XLS Padlock afin que toutes les modifications récentes soient incluses dans la compilation.

3. Générez votre application

Après avoir configuré les paramètres de votre application, cliquez sur le bouton « Build Secure Application » ou appuyez sur **F5**. XLS Padlock compilera votre classeur et créera le fichier EXE de l'application.



Une fois l'opération terminée, un message de confirmation apparaîtra :



Depuis cette boîte de dialogue, vous pouvez immédiatement exécuter l'application, consulter le journal de compilation (particulièrement utile pour résoudre les erreurs) ou ouvrir le dossier de destination.

CONSEIL

XLS Padlock ne modifie pas votre fichier de classeur Excel d'origine pendant la compilation. Conservez toujours une sauvegarde de votre classeur d'origine.

Résolution des problèmes de mise en forme

Si votre classeur protégé affiche des résultats étranges ou des problèmes de mise en forme, essayez d'activer l'option « [Use Excel automation for formula protection](#) » sur la page « Formulas and Passwords ».

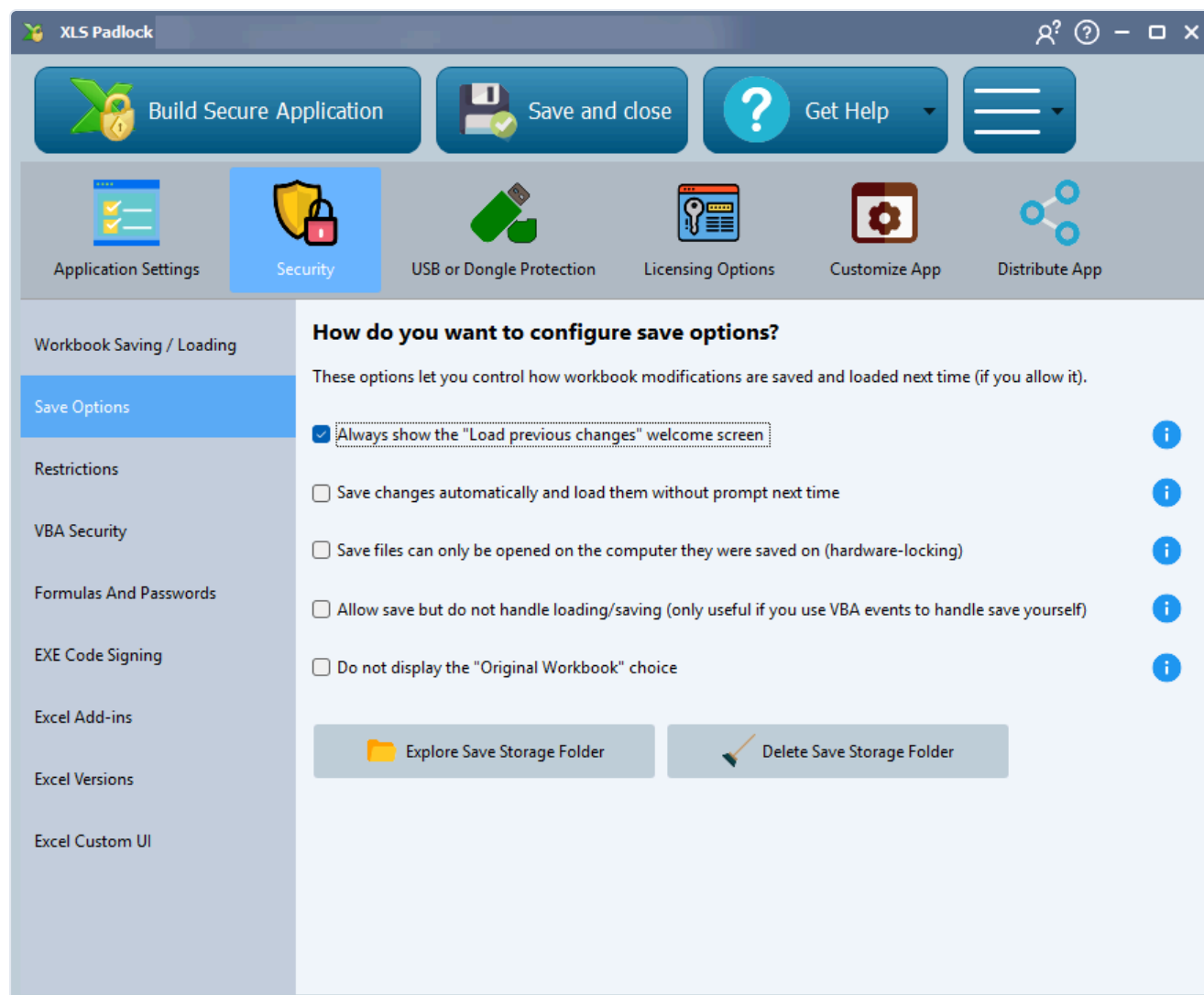
👉 Voir aussi :

[Comment distribuer un classeur protégé](#)

Sécurité

XLS Padlock offre de nombreuses façons de protéger votre classeur Excel, notamment des options d'enregistrement, des restrictions de fonctionnalités et des conditions d'ouverture de votre application, comme l'exigence d'une [clé d'activation](#) ou la [présence d'un dongle](#) ou d'une [clé USB](#).

Cette rubrique présente un aperçu des options de sécurité disponibles sous « **Security** » dans XLS Padlock.



Workbook Saving / Loading

Configurez la possibilité pour les utilisateurs finaux d'enregistrer les modifications apportées au classeur protégé. Lorsque l'enregistrement est activé, XLS Padlock crée des fichiers sécurisés `.XLSC` ou `.XLSCE` qui ne peuvent être ouverts que par votre application, protégeant ainsi les données de l'utilisateur.

[→ En savoir plus sur l'enregistrement et le chargement du classeur](#)

Save Options

Affinez le comportement d'enregistrement avec des options telles que l'enregistrement automatique des modifications à la fermeture, le verrouillage matériel des fichiers de sauvegarde sur un ordinateur spécifique, ou l'affichage permanent de l'écran d'accueil pour permettre aux utilisateurs de choisir entre repartir de zéro et charger un travail précédent.

→ [Découvrir les options d'enregistrement avancées](#)

Restrictions

Restreignez les fonctionnalités clés d'Excel pour renforcer la sécurité. Vous pouvez désactiver le menu contextuel du clic droit, la barre de formule, les commandes copier/coller, l'impression, et bien plus encore, afin d'empêcher les utilisateurs d'accéder aux parties sensibles de votre classeur ou de les modifier.

→ [Voir les fonctionnalités de restriction](#)

VBA Security

Protégez votre code VBA contre toute consultation ou modification. Vous pouvez verrouiller le projet VBA avec un mot de passe masqué pour l'utilisateur ou bloquer complètement l'accès à l'éditeur VBA (VBE), rendant inutiles les outils de craquage de mots de passe. Pour une protection optimale, utilisez le VBA Compiler.

→ [Voir les différents moyens de protection VBA](#)

Formulas and Passwords

Sécurisez vos formules et la structure de vos feuilles de calcul. Utilisez la protection des formules de XLS Padlock pour permettre aux utilisateurs d'interagir avec les formules sans pouvoir les voir ni les copier. Vous pouvez également gérer les mots de passe du classeur et des feuilles de calcul directement depuis l'interface.

→ [En savoir plus sur la protection des formules](#)

EXE Code Signing

Signez numériquement votre fichier `.EXE` compilé pour instaurer la confiance auprès de vos utilisateurs et éviter les avertissements de Windows SmartScreen et des logiciels antivirus. XLS Padlock peut automatiser le processus de signature si vous disposez d'un certificat de signature de code.

→ [Comment signer numériquement votre application](#)

Excel Add-ins

Par défaut, XLS Padlock désactive la plupart des compléments par mesure de sécurité. Cette section vous permet de réactiver de manière sélective des compléments COM spécifiques ou des compléments Excel courants (comme l'Analysis ToolPak ou le Solver) nécessaires au bon fonctionnement de votre classeur.

→ [Comment gérer les compléments Excel](#)

Excel Versions

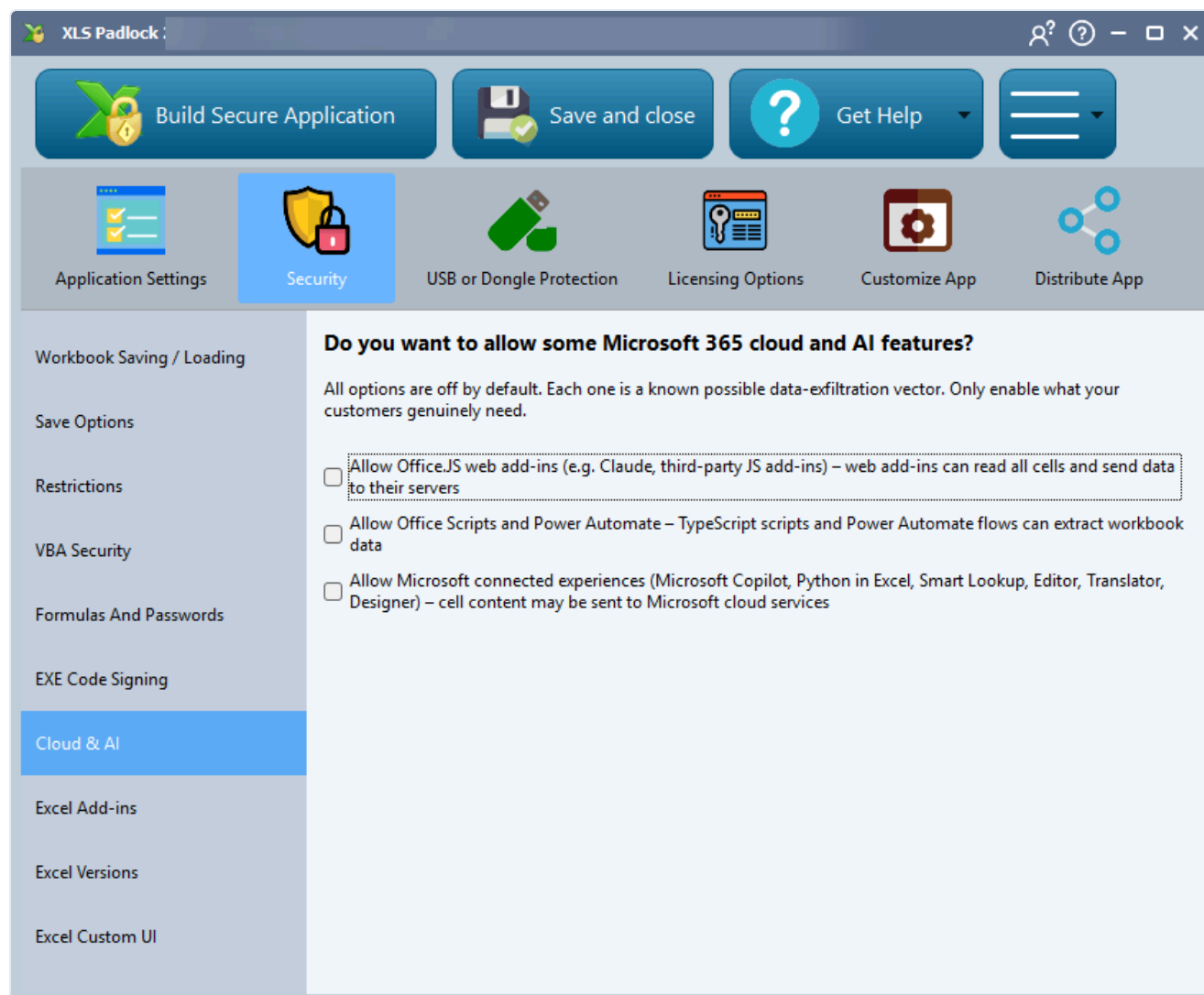
Spécifiez quelles versions de Microsoft Excel sont compatibles avec votre application protégée. Si un utilisateur tente d'exécuter l'EXE sur une version non prise en charge, un message d'erreur personnalisable s'affichera.

[→ Comment exiger des versions spécifiques d'Excel](#)

Options Cloud et IA

Excel moderne intègre plusieurs fonctionnalités cloud et IA capables de lire votre classeur et d'envoyer son contenu vers des serveurs externes. Pour garder votre application protégée hermétique, XLS Padlock les bloque toutes par défaut. C'est vous qui décidez, une par une, lesquelles autoriser.

Vous trouverez ces paramètres dans le concepteur XLS Padlock, sous l'onglet **Security**, sur la page **Cloud & AI**.



Pourquoi ces fonctionnalités sont bloquées par défaut

Chaque option de cette page est un moyen connu pour les données du classeur de quitter la machine de l'utilisateur final. Un classeur protégé est censé garder vos formules, vos données et votre code VBA sous votre contrôle. XLS Padlock part donc de la position la plus sûre : tout est désactivé. N'activez que ce dont vos clients ont réellement besoin.

Options disponibles

Les trois options sont décochées (bloquées) par défaut.

Allow Office JS web add-ins

Les compléments web Office JS (aussi appelés web add-ins) s'exécutent à l'intérieur d'Excel et peuvent lire chaque cellule du classeur ouvert, puis envoyer ces données vers leurs propres serveurs. Cela inclut les compléments tiers tels que le plugin **Claude for Excel** ainsi que tout autre complément JS installé depuis l'Office store ou chargé de manière indépendante (sideloaded).

Lorsque cette option est désactivée, les compléments web ne peuvent pas se charger contre votre classeur protégé. Ne l'activez que si votre application dépend d'un complément web précis auquel vous faites confiance.

Allow Office Scripts and Power Automate

Les Office Scripts (scripts d'automatisation TypeScript) et les flux Power Automate peuvent lire le contenu du classeur et le sortir de l'application. Laissez cette option désactivée, sauf si votre flux de travail en dépend.

Allow Microsoft connected experiences

Les expériences connectées Microsoft (Microsoft connected experiences) envoient le contenu des cellules vers les services cloud de Microsoft. Cela couvre Microsoft Copilot, Python dans Excel, la Recherche intelligente (Smart Lookup), l'Éditeur (Editor), le Traducteur (Translator) et le Concepteur (Designer). Ne l'activez que si vos clients ont besoin de ces fonctionnalités et que vous acceptez que le contenu des cellules concernées puisse être traité dans le cloud Microsoft.

Remarques

- Ces options s'appliquent à l'application compilée, pas au concepteur XLS Padlock.
- Le blocage de ces surfaces masque également leurs points d'entrée dans le ruban Excel (par exemple le menu déroulant des compléments, l'onglet Automate et le bouton Copilot), de sorte que les utilisateurs finaux ne sont pas invités à utiliser une fonctionnalité qui est désactivée.
- Les compléments COM et les compléments Excel intégrés sont gérés séparément. Voir [Comment gérer les compléments Excel](#).

Améliorer la protection

Pourquoi compiler en EXE ne suffit pas

Lorsque vous compilez un classeur en une application protégée, le fichier classeur d'origine (XLSX, XLSM, etc.) est chiffré et intégré dans le fichier .EXE résultant. Cela empêche les utilisateurs d'accéder directement au fichier d'origine.

La protection de base ne suffit pas

Cependant, le simple fait de compiler votre classeur ne fournit qu'une couche de sécurité élémentaire. Voyez cela comme fermer une porte sans la verrouiller. Pour réellement sécuriser votre propriété intellectuelle, vous devez activer les fonctionnalités de protection avancées offertes par XLS Padlock.

Pour fonctionner, XLS Padlock doit charger votre classeur dans la mémoire d'Excel. **Sans activer les couches de sécurité supplémentaires de XLS Padlock**, un attaquant sophistiqué pourrait potentiellement extraire le classeur d'origine de la mémoire pendant que l'application est en cours d'exécution. Bien que ce ne soit pas une tâche triviale, cela souligne le besoin d'une protection plus forte.

Rendre un classeur extrait inutilisable

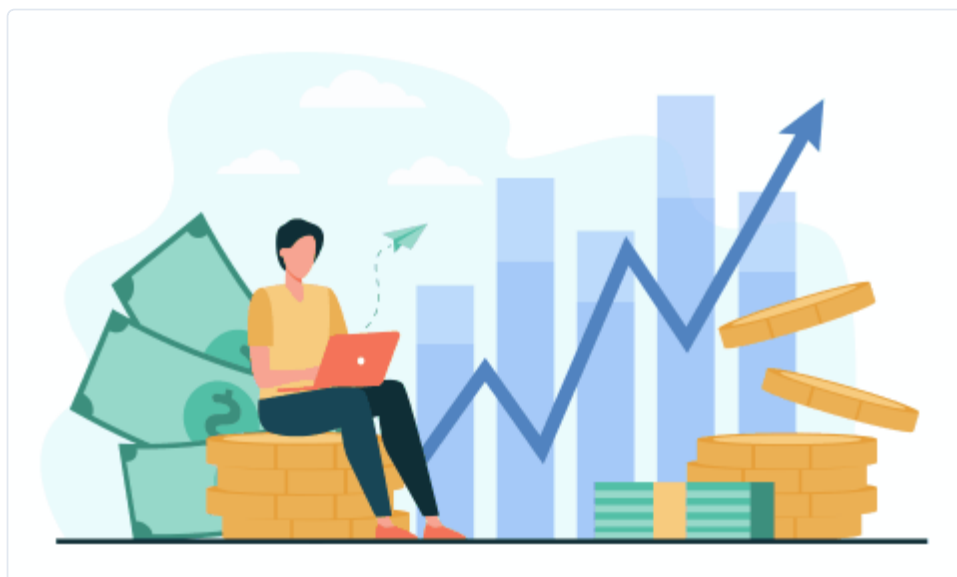
Lorsque vous appliquez correctement les fonctionnalités de sécurité de XLS Padlock, le classeur extrait devient inutilisable. Même si quelqu'un extrait le fichier, il ne fonctionnera pas correctement. La protection est particulièrement forte pour les classeurs contenant des formules et/ou du code VBA, car ces composants critiques sont soit chiffrés, soit compilés, ce qui empêche efficacement toute utilisation non autorisée ou toute altération.

Les fonctionnalités de sécurité les plus efficaces

Nous recommandons d'activer les fonctionnalités suivantes pour la protection la plus forte :

- [Protégez vos formules avec la protection des formules](#)
- [Sécurisez votre VBA avec le VBA Compiler](#)
- [Prévenez les piratages VBA et OLE courants](#)
- [Interdisez l'accès à l'éditeur VBA \(VBE\)](#)
- [Protégez votre classeur par mot de passe](#)
- [Désactivez les compléments Excel inutiles](#)

Contrôle d'accès au classeur



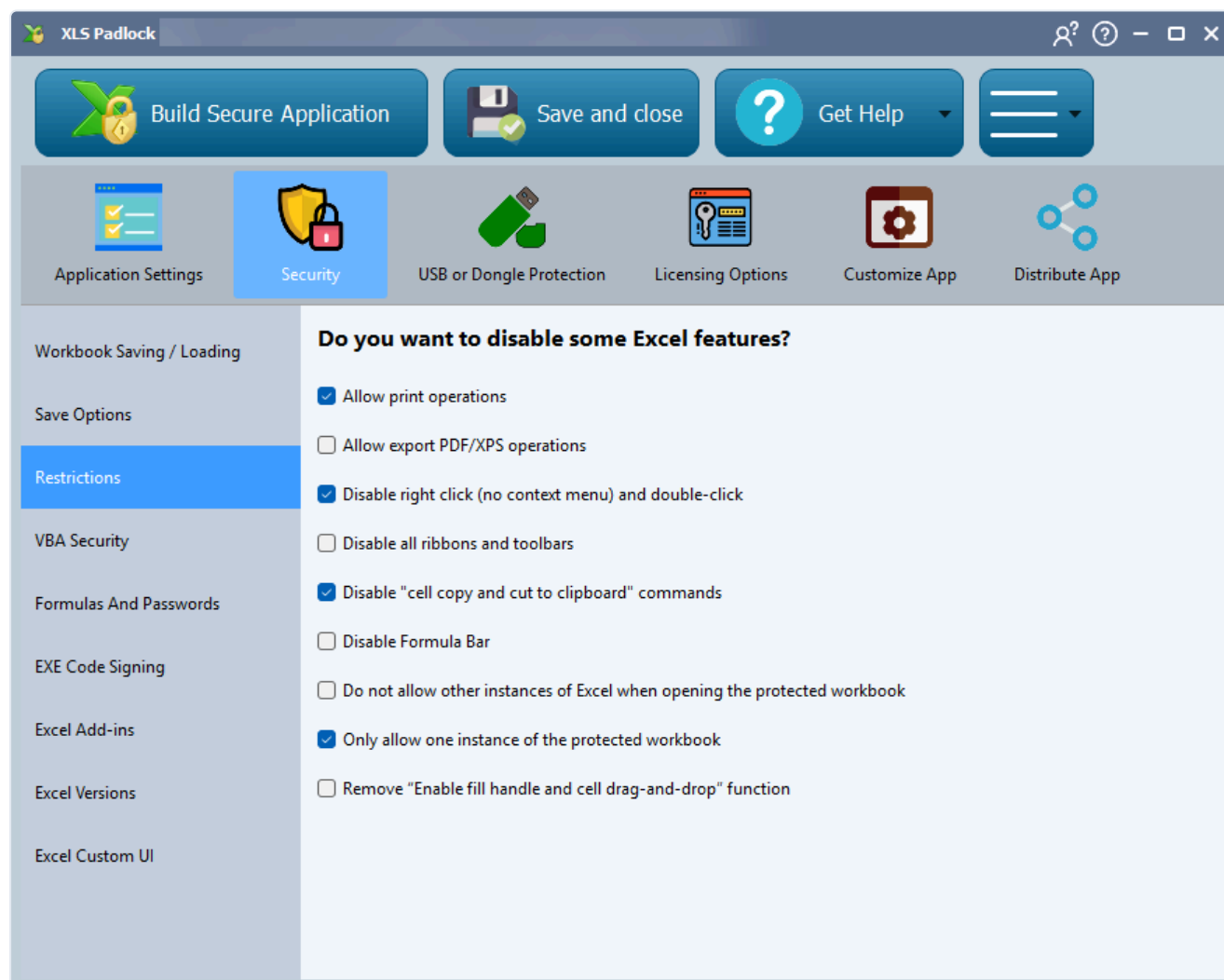
XLS Padlock fournit de nombreuses fonctionnalités de sécurité pour **restreindre l'accès à vos classeurs Excel aux seuls utilisateurs autorisés**. Vous pouvez générer des clés d'activation, définir des essais à durée limitée, et bien plus encore. Un objectif principal de XLS Padlock est de vous permettre de **vendre des licences pour vos classeurs**, en fournissant des fonctionnalités pour réduire le piratage et le partage non autorisé.

Ce sujet donne un aperçu des fonctionnalités de sécurité dédiées au contrôle d'accès aux classeurs :

- **Contrôlez qui peut utiliser votre classeur** : [Configurez des clés d'activation](#).
- **Empêchez le partage en verrouillant une licence sur un seul ordinateur** : Utilisez les [clés d'activation verrouillées au matériel](#).
- **Gérez les activations à distance et désactivez l'accès** : Utilisez l'[activation en ligne](#), la [désactivation](#) et la [validation](#).
- **Créez des versions d'essai de votre classeur** : Apprenez [comment créer des classeurs d'essai](#).
- **Restreignez les actions des utilisateurs** : [Interdisez l'impression, l'exportation, le clic droit, et plus encore](#).
- **Exigez un périphérique physique pour l'accès** : Liez l'application à un [dongle ou une clé USB](#).
- **Vendez des licences et des abonnements en ligne** : Intégrez des plateformes e-commerce comme [WooCommerce](#) ou [FastSpring](#).

Restrictions du classeur Excel

Ce sujet traite des options disponibles dans la section 'Security > Restrictions' de XLS Padlock, qui vous permettent de restreindre certaines actions de l'utilisateur dans votre classeur protégé.



Allow Print Operations

Pour empêcher les utilisateurs d'imprimer votre classeur, désactivez l'option **Allow print operations** (Autoriser les opérations d'impression) dans les paramètres de [Security](#).

Si cette option est désactivée, toute tentative d'impression sera bloquée, et l'utilisateur verra un message d'erreur "Printing is not allowed".

Allow Export to PDF/XPS

Pour empêcher les utilisateurs d'exporter le contenu du classeur vers des fichiers PDF ou XPS, désactivez l'option "Allow export PDF/XPS operations" dans les paramètres de [Security](#).

Disable Right-Click (and optionally Double-Click)

Pour désactiver le menu contextuel du clic droit sur les cellules de votre classeur, activez l'option "Disable right-click" dans les paramètres de [Security](#).

Ceci est couramment utilisé pour empêcher les utilisateurs de copier, coller ou modifier les données et la mise en forme des cellules.

Disable All Ribbons and Toolbars

L'option **Disable all ribbons and toolbars** (Désactiver tous les rubans et barres d'outils) masque tous les onglets de ruban intégrés et la plupart des commandes du menu Microsoft Office, créant ainsi une interface simplifiée de type kiosque pour votre application.

Cette fonctionnalité fonctionne avec **Excel 2007 et versions ultérieures**. Vous pouvez l'imposer en définissant la [version minimale d'Excel requise](#) pour votre classeur.

Disable "Cell Copy and Cut to Clipboard" Commands

Activez cette option pour désactiver les commandes **Copy** et **Cut** pour les cellules. Cela empêche les utilisateurs finaux de copier facilement les données de votre classeur vers le presse-papiers pour les réutiliser dans d'autres applications.

Disable Formula Bar

L'option **Disable Formula Bar** (Désactiver la barre de formule) masque la barre de formule d'Excel. De plus, l'option correspondante dans le menu Affichage d'Excel est désactivée, de sorte que les utilisateurs finaux ne peuvent pas la réactiver.

Pour une sécurité renforcée, nous vous recommandons également de [désactiver l'accès à l'éditeur VBA](#).

Do Not Allow Other Instances of Excel

Lorsqu'elle est activée, cette option vérifie si une autre instance d'Excel est déjà en cours d'exécution lorsqu'un utilisateur démarre votre classeur compilé. Si un processus Excel existant est détecté, un message invitera l'utilisateur à le fermer avant de continuer.

Cela garantit que l'instance d'Excel exécutant votre classeur protégé est la première et la seule, offrant ainsi un environnement plus contrôlé.

Cette option se trouve dans les paramètres de [Security](#).

Only Allow One Instance of the Protected Workbook

L'option "Only allow one instance of the protected workbook", située dans les paramètres de [Security](#), empêche les utilisateurs finaux d'exécuter plus d'une instance de votre application protégée en même temps. Si un utilisateur tente d'ouvrir à nouveau l'application alors qu'elle est déjà en cours d'exécution, la fenêtre existante sera ramenée au premier plan. Cette fonctionnalité empêche efficacement les utilisateurs d'ouvrir plusieurs instances du même classeur protégé.

Note de compatibilité

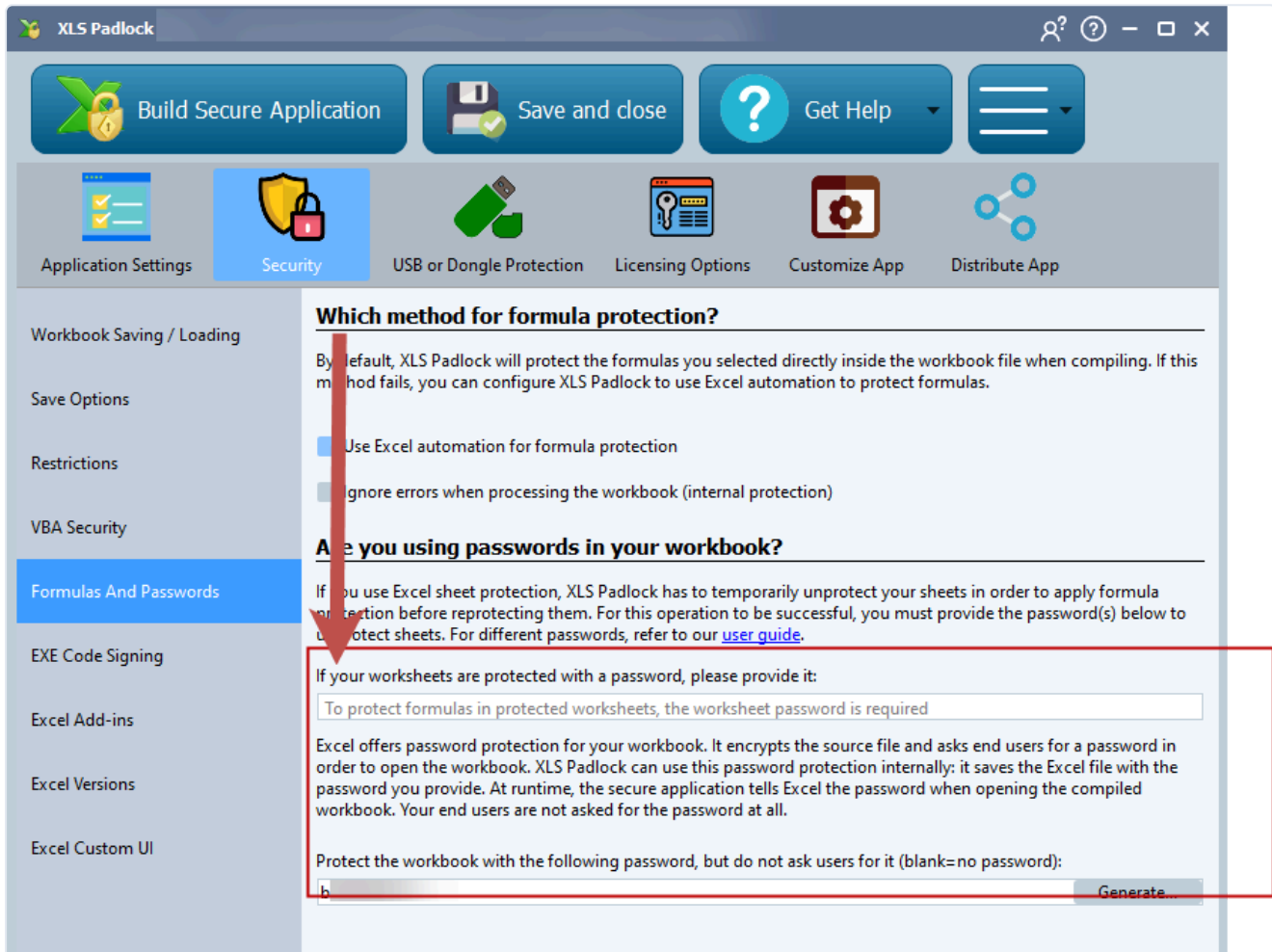
Sur les versions d'Excel antérieures à 2013, une boîte de message indiquant "An instance of this application is already running" sera affichée au lieu de mettre le focus sur la fenêtre existante.

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

Cette option "Remove "Enable fill handle and cell drag-and-drop" function" empêche les utilisateurs finaux d'utiliser la poignée de recopie et le glisser-déposer de cellules dans le classeur protégé. Elle est identique à celle des options générales d'Excel (Review", "Check accessibility", "Options accessibility", "Advanced", "Enable fill handle and cell drag-and-drop"), mais elle vous permet de remplacer le choix de l'utilisateur local.

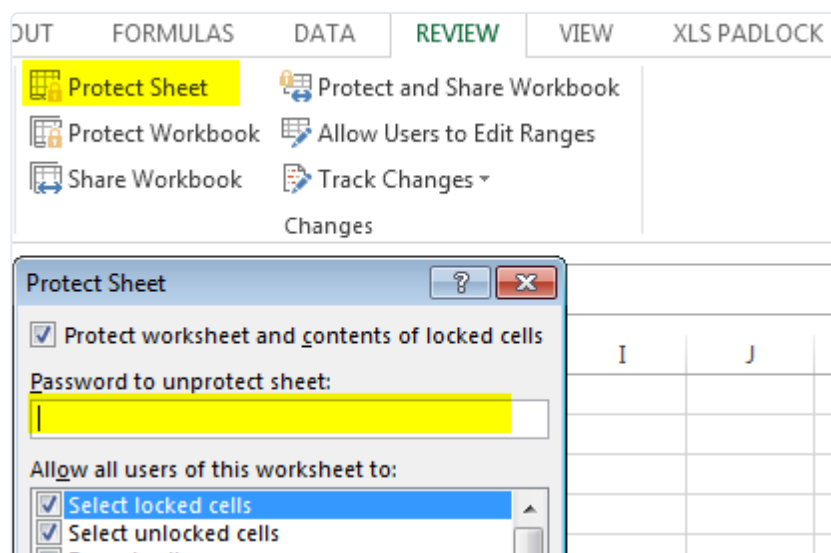
Protection par mot de passe des feuilles

XLS Padlock vous permet de gérer à la fois les mots de passe de classeur et de feuille de calcul afin de renforcer la sécurité et les fonctionnalités de votre application protégée. Ce guide explique comment configurer ces paramètres de mot de passe.



Protection par mot de passe des feuilles de calcul

Si vous utilisez la protection de feuille intégrée d'Excel pour restreindre l'accès aux cellules, XLS Padlock doit connaître votre mot de passe pour gérer ses propres couches de sécurité.



Lorsque la [Protection des formules](#) de XLS Padlock est active, il doit temporairement déprotéger vos feuilles, appliquer sa protection, puis les protéger à nouveau. Pour réaliser cette opération avec succès, vous devez fournir le mot de passe utilisé pour la protection de feuille dans Excel.

REMARQUE

XLS Padlock réapplique automatiquement les mêmes options de protection de feuille que celles définies dans le classeur d'origine.

Si vous utilisez un mot de passe unique pour toutes les feuilles de calcul protégées, saisissez-le dans le champ "If your worksheets are protected with a password, please provide it".

Si vous utilisez des mots de passe différents pour différentes feuilles, vous devez les fournir en utilisant le format JSON suivant :

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

Par exemple :

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

Protection silencieuse par mot de passe du classeur

Excel offre une fonctionnalité permettant de chiffrer l'intégralité du classeur avec un mot de passe. XLS Padlock peut gérer ce mot de passe pour vous, offrant une couche de sécurité supplémentaire sans solliciter vos utilisateurs finaux.

Lorsque vous définissez un mot de passe dans le champ "Protect the workbook with the following password", XLS Padlock enregistre le classeur avec ce mot de passe. Au moment de l'exécution, l'application sécurisée fournit automatiquement le mot de passe à Excel lors de l'ouverture du classeur.

Vos utilisateurs finaux ne se voient jamais demander le mot de passe.

Cette combinaison augmente la sécurité de votre application en superposant la protection de XLS Padlock au chiffrement natif d'Excel.

Vous pouvez utiliser le bouton **Generate** pour créer un mot de passe fort et aléatoire.

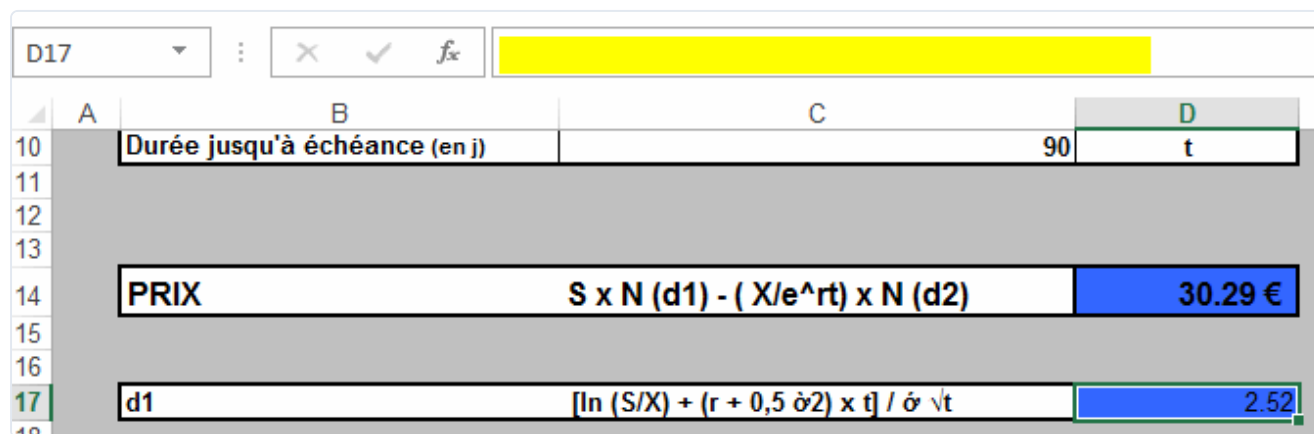
Ne modifiez pas le mot de passe après distribution

Une fois que vous avez distribué une application protégée par un mot de passe de classeur spécifique, ne le modifiez pas. Si vous le faites, les utilisateurs finaux ne pourront pas ouvrir leurs données précédemment enregistrées, car l'application sera incapable de les déchiffrer.

Si vous laissez le champ du mot de passe vide, cette fonctionnalité sera désactivée.

Combiner la protection Excel et XLS Padlock

La protection des formules de XLS Padlock est entièrement compatible avec les fonctionnalités de protection de feuille intégrées d'Excel, telles que les cellules verrouillées ou masquées. Lorsque vous utilisez les deux, les cellules protégées n'afficheront rien dans la barre de formule, et les appels de fonction `PLEvalForm` sous-jacents resteront masqués.



The screenshot shows an Excel spreadsheet with 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

Avertissement sur les dépendances

Les formules protégées qui dépendent d'autres cellules échoueront si ces cellules sont masquées à l'aide de l'attribut de format de cellule "Hidden" d'Excel. Évitez d'utiliser l'attribut masqué sur les cellules qui sont des dépendances de vos formules protégées.

Voir aussi :

- [Désactiver la protection des formules](#)

Protection des formules : Excel ou XLS Padlock

Bien qu'Excel vous permette de masquer les formules sur une feuille protégée, cette protection est limitée. XLS Padlock va beaucoup plus loin en remplaçant vos formules par une fonction sécurisée qui n'est disponible que lorsque votre application est en cours d'exécution.

Dans le classeur compilé, les formules sont remplacées dans la barre de formule par des fonctions anonymes telles que `=PLEvalFormD(0;0)` ou `=PLEvalFormD(1;COUNT($F7))`. Les calculs continueront cependant de fonctionner comme prévu.

Avant la protection, n'importe qui peut voir votre formule :

The screenshot shows an Excel spreadsheet with the following data:

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

The formula bar for cell D17 contains: `=LN(C6/C7)+(C9+0.5*C8^2)*(C10/360))/(C8*(C10/360)^(1/2))`

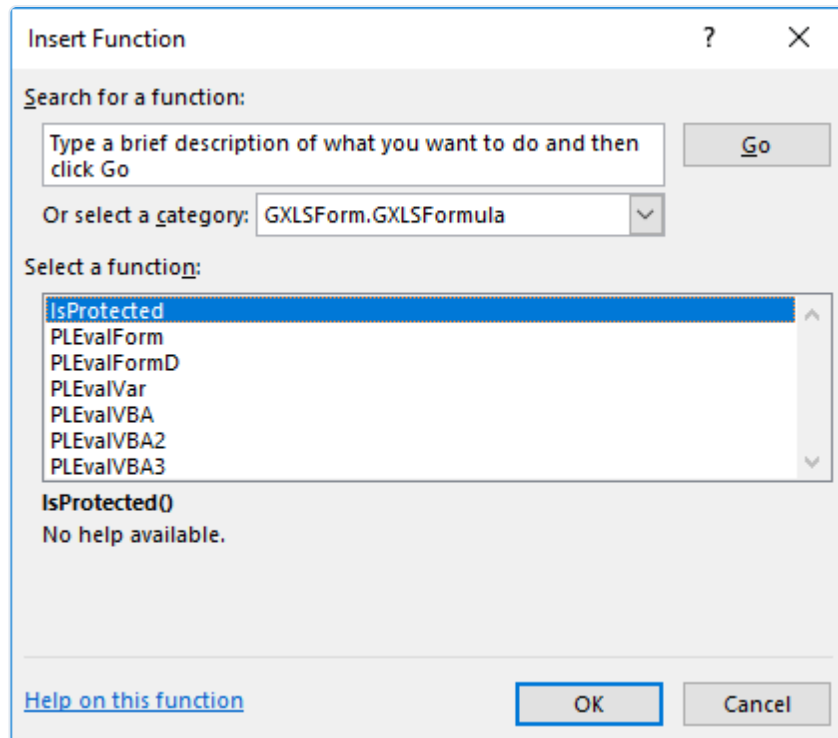
Après la protection avec XLS Padlock, la formule est automatiquement remplacée tout en restant fonctionnelle :

The screenshot shows the same Excel spreadsheet after protection. The formula bar for cell D17 now contains: `=XLSPadlockForm(0)`

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

Cela augmente considérablement la sécurité de votre classeur. Comme les formules originales sont supprimées du fichier du classeur, les cellules protégées ne peuvent fonctionner correctement que lorsque le classeur est ouvert au sein de l'application sécurisée créée par XLS Padlock.

Même au moment de la conception, vous pouvez constater que les fonctions de sécurité de XLS Padlock comme `PLEvalForm` et `PLEvalFormD` sont reconnues par Excel :

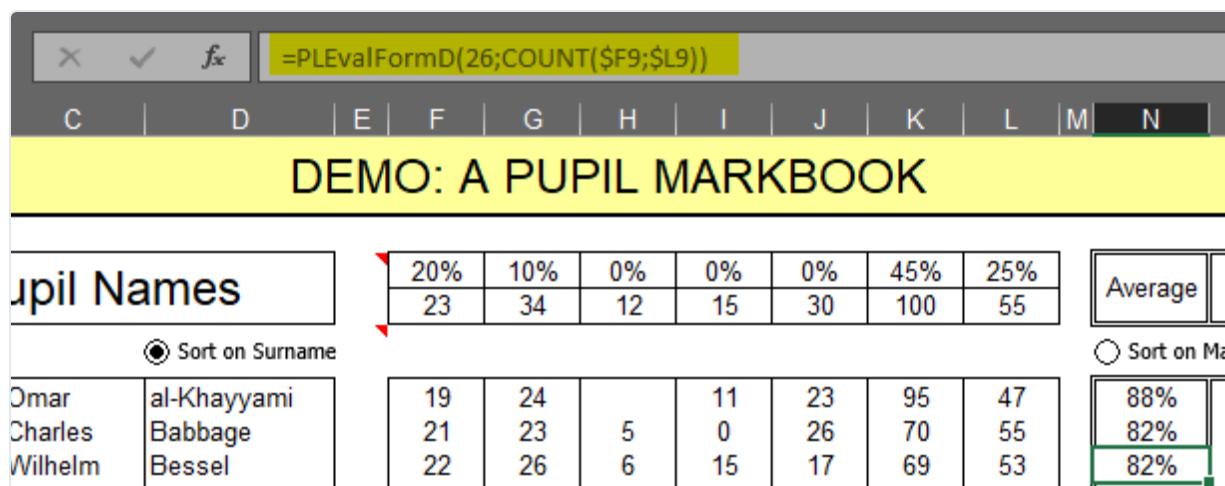


👉 En savoir plus sur la [protection des cellules avec XLS Padlock](#).

Protéger les formules avec XLS Padlock

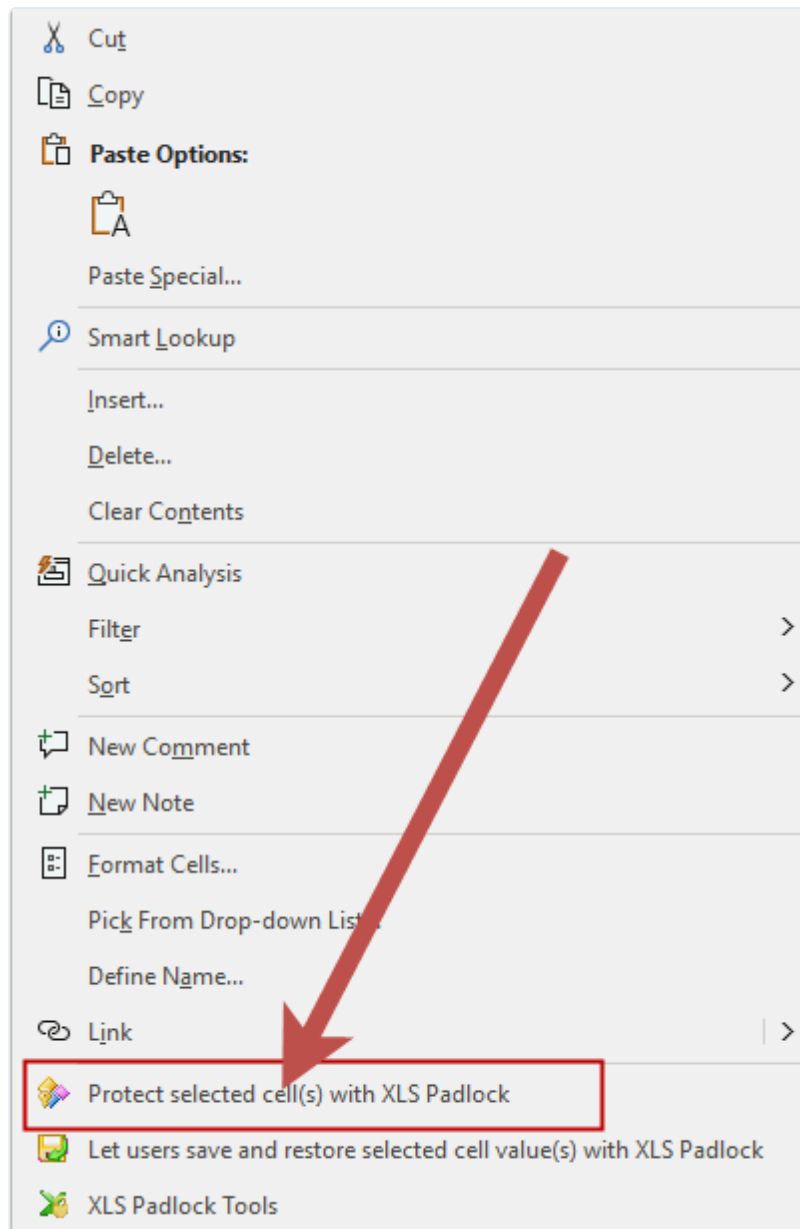
En plus de la protection des cellules d'Excel, **nous vous recommandons vivement de sécuriser les formules importantes avec la protection des formules propre à XLS Padlock.**

Après la protection avec XLS Padlock, vos formules n'apparaîtront plus dans la barre de formule, **tout en restant pleinement fonctionnelles.** À la place, un appel de fonction cryptique tel que `PLEvalFormD` est affiché :



👉 L'objectif est de rendre le classeur inutilisable si quelqu'un parvient à [extraire le fichier du classeur de l'EXE compilé](#). Comme les **formules protégées par XLS Padlock ne se trouvent plus dans le fichier du classeur**, le classeur extrait ne fonctionnera pas correctement.

XLS Padlock vous laisse décider quelles cellules protéger, et vous pouvez **sélectionner et protéger plusieurs cellules à la fois**.



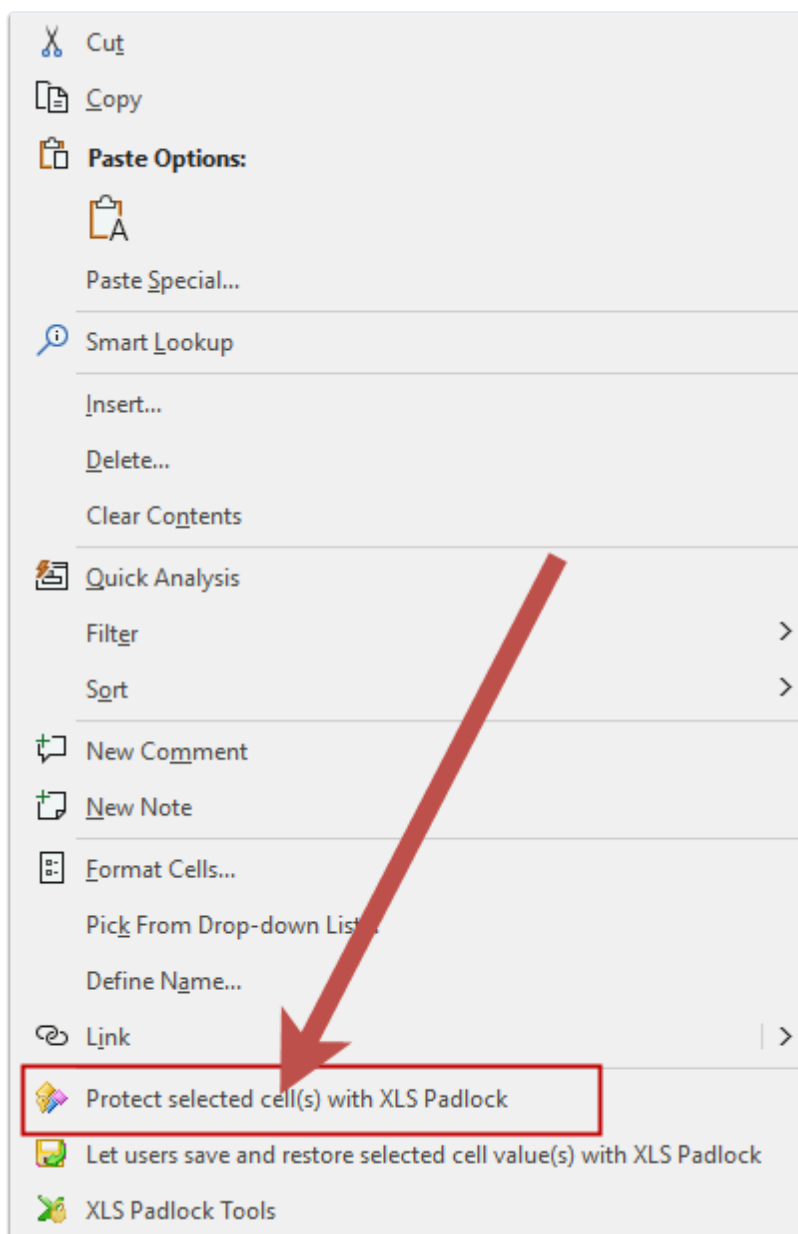
👉 Voir comment sélectionner les cellules à protéger

Voir aussi :

- [En savoir plus sur comment protéger vos formules efficacement](#)
- [Désactiver les compléments Excel courants](#)
- [Utiliser le VBA Compiler pour une protection encore plus forte](#)

Marquer les cellules à protéger

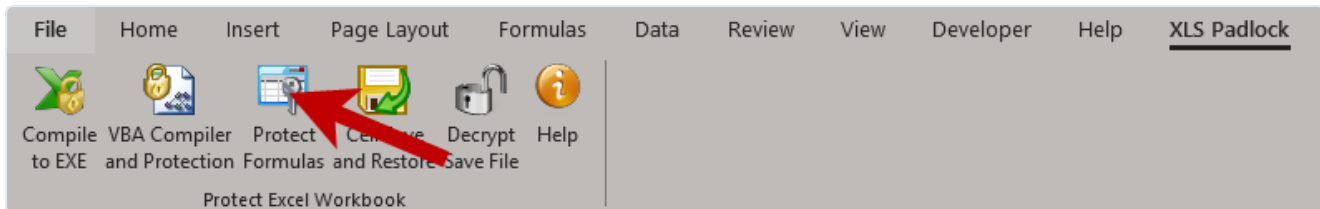
Pour protéger une formule, faites simplement un **clic droit sur une ou plusieurs cellules** contenant les formules que vous souhaitez masquer et sélectionnez « **Protect selected cell(s) with XLS Padlock** » dans le menu contextuel.



XLS Padlock confirmera alors que les cellules sont marquées pour la protection.

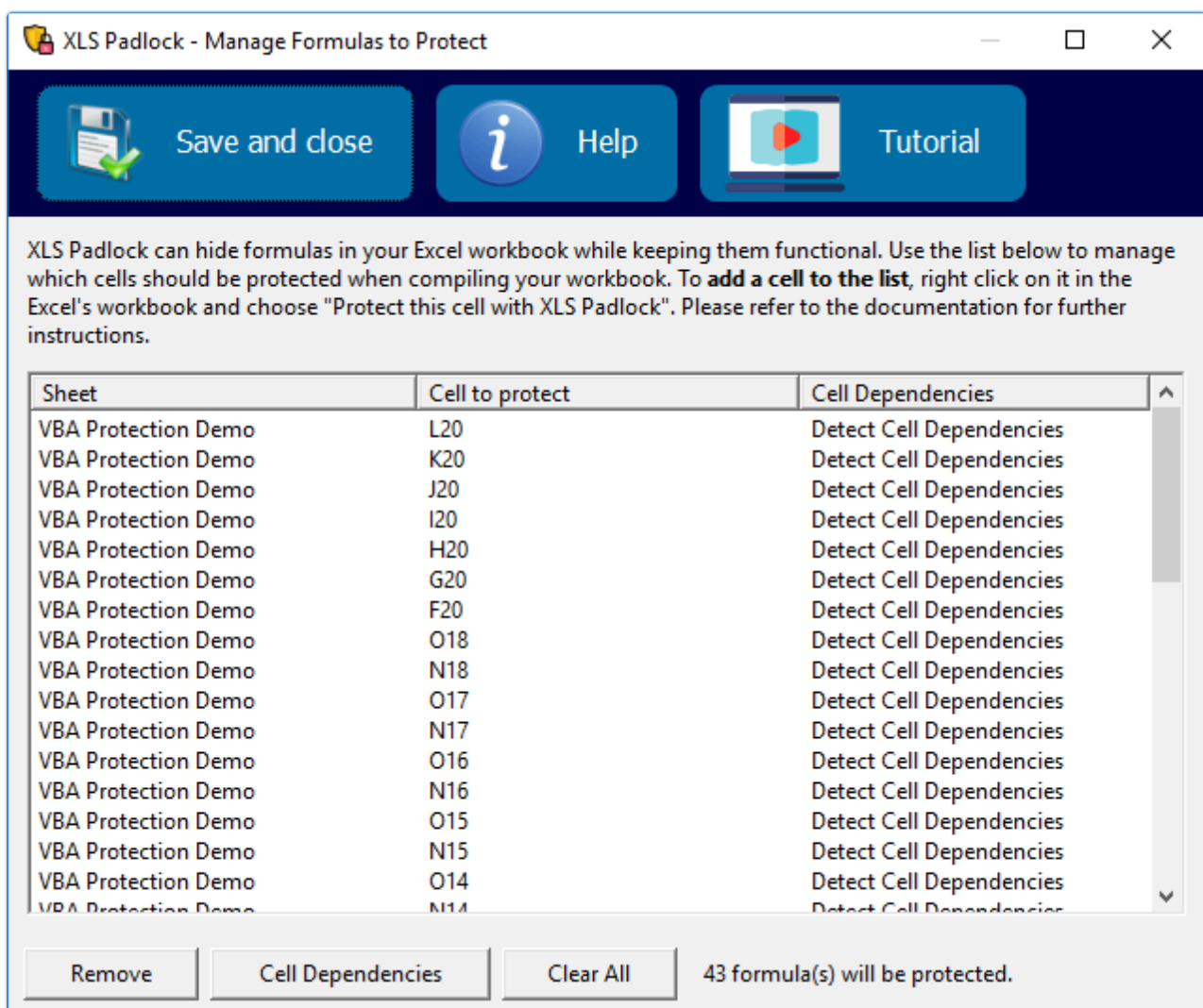
Vue d'ensemble des cellules protégées

Pour voir une vue d'ensemble de toutes les cellules protégées, cliquez sur « **Protect Formulas** » dans l'onglet ou le menu XLS Padlock :



Cela ouvre une liste de toutes les cellules configurées pour la protection. Dans cette fenêtre, vous pouvez ajuster le comportement de la protection, retirer la protection de cellules spécifiques ou effacer la liste entière.

Lorsque vous compilez votre classeur, XLS Padlock remplace toutes les formules listées par des appels de fonction génériques tels que `PLEvalForm(N)` et `PLEvalFormD(N, ...)`. **Vos cellules restent fonctionnelles, mais les utilisateurs finaux ne peuvent pas découvrir les formules sous-jacentes.** Les formules originales n'existent plus dans le classeur compilé ; elles sont gérées par l'EXE lui-même.



Dépendances de cellules

Le bouton **Cell Dependencies** vous permet de contrôler comment la protection est appliquée. Deux choix sont disponibles : « Detect Cell Dependencies » et « No ».

Par défaut, XLS Padlock détectera toutes les références de cellules et les noms de plages dans vos formules (dépendances de cellules) et générera une fonction anonyme qui contient ces références. Cela permet à Excel de recalculer correctement les cellules protégées. Par exemple, si la formule à protéger est `=A3^2`, XLS Padlock générera une fonction telle que : `PLEvalFormD(1, COUNT(A3))`.

Si XLS Padlock ne parvient pas à protéger une cellule, vous pouvez choisir « **No** » pour son réglage « Cell Dependencies ». Dans ce cas, une simple fonction générique `PLEvalForm(N)` sera utilisée.

Limites de la protection

Certaines formules complexes ne sont pas prises en charge par la protection et échoueront, affichant un `#ERROR!` ou `#VALUE!` dans la cellule. Veuillez tester vos formules en profondeur avant de distribuer votre classeur protégé. Portez une attention particulière aux limites suivantes :

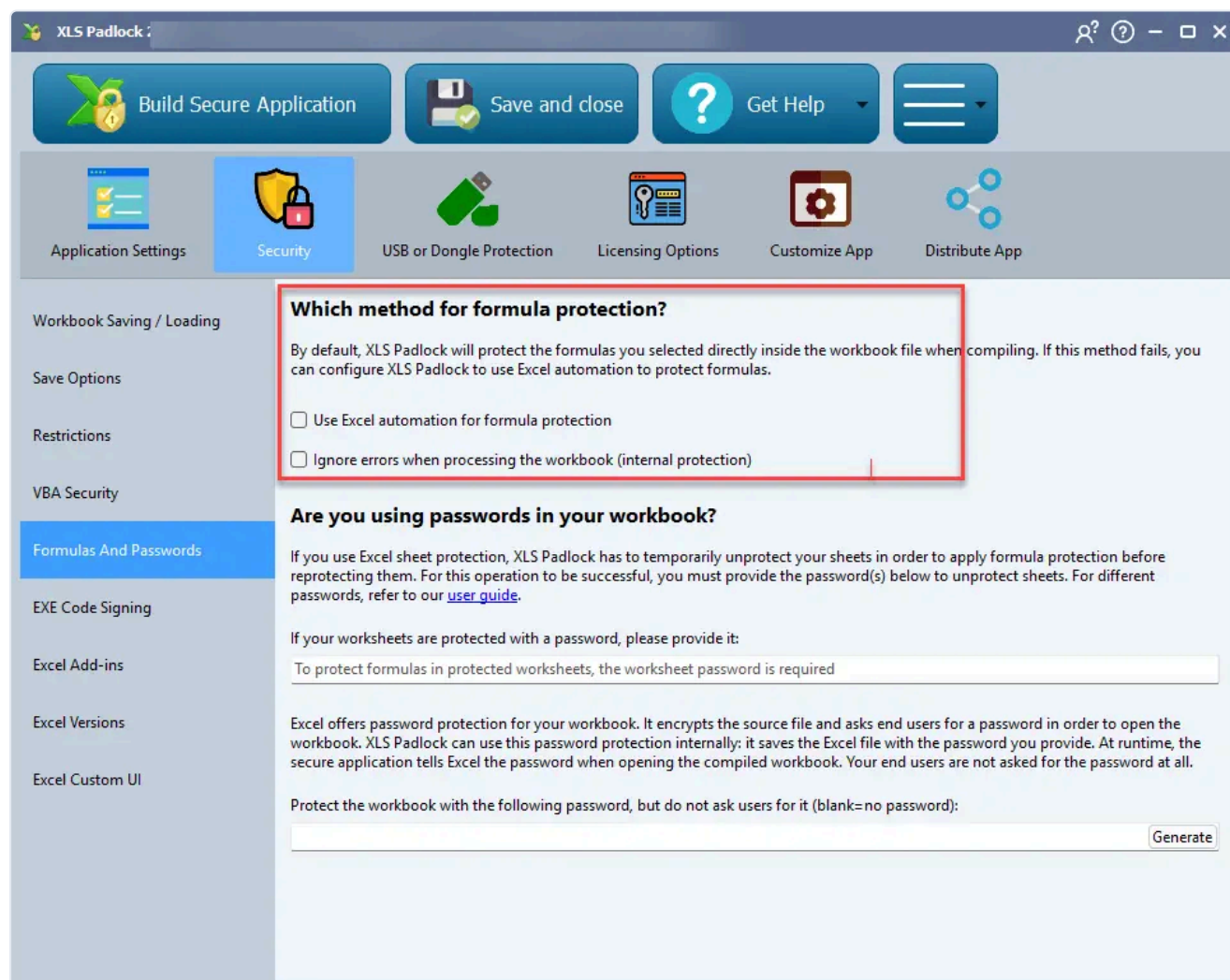
- La longueur de la formule doit être inférieure à 256 caractères.
- La fonction `INDIRECT` n'est pas prise en charge.
- Les formules ne doivent contenir aucune fonction VBA définie par l'utilisateur, uniquement des fonctions Excel classiques.
- La protection des formules doit être utilisée avec parcimonie. Ne protégez que les formules les plus importantes dont dépend votre classeur. Protéger plusieurs milliers de formules n'est pas recommandé, car cela augmentera la taille de votre fichier EXE et peut ralentir Excel sur les ordinateurs plus anciens.
- **La validation des données de cellule et les règles de mise en forme conditionnelle ne sont pas prises en charge.**

👉 Voir aussi

- [Combiner la protection de feuille Excel et la protection XLS Padlock](#)

Méthode de protection des formules

Lorsque vous activez la protection des formules, XLS Padlock peut utiliser l'une des deux méthodes suivantes :



1. **Direct Modification (Recommended)** (modification directe, recommandée) : cette méthode modifie les formules directement dans le fichier source du classeur. C'est la méthode par défaut et elle fonctionne de manière fiable dans la plupart des cas.
2. **Excel Automation** (automatisation Excel) : cette méthode utilise l'automatisation pour piloter Excel et appliquer la protection.

Si la méthode recommandée échoue, vous pouvez forcer XLS Padlock à utiliser la méthode d'automatisation en activant l'option "**Use Excel automation for formula protection**". Cette méthode est également utilisée automatiquement pour les classeurs binaires (XLSB) et les très gros fichiers.

En dernier recours, vous pouvez activer "[Ignore errors when processing the workbook](#)", mais veuillez à tester minutieusement votre application compilée si vous activez cette option.

Désactiver la protection des formules

Étant donné que la protection des formules de XLS Padlock est incompatible avec Microsoft Excel 2000, vous devez activer l'option **Disable formula and VBA protection** sur la page Security si vous devez créer des applications compatibles avec cette version.

Le VBA Compiler est également désactivé

L'activation de cette option désactivera également le compilateur de code VBA.

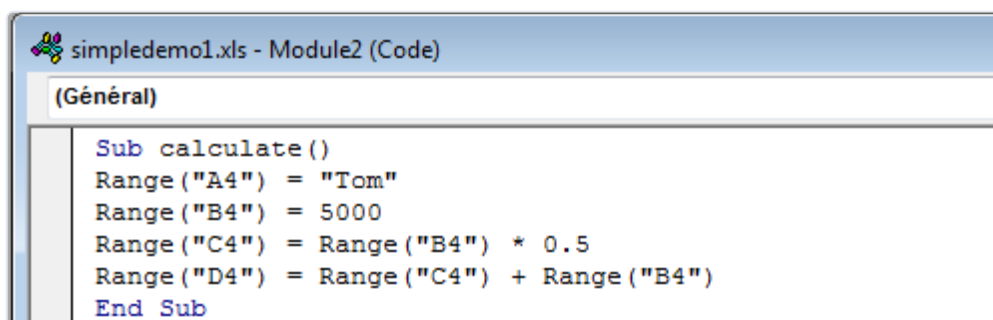
À propos du VBA Compiler intégré

XLS Padlock dispose d'un **compilateur VBA intégré** qui transforme vos scripts Basic en bytecode sécurisé, les rendant inaccessibles aux utilisateurs finaux. La compilation de parties de vos macros VBA supprime le code source d'origine, le rendant impossible à copier.

Tutoriel vidéo

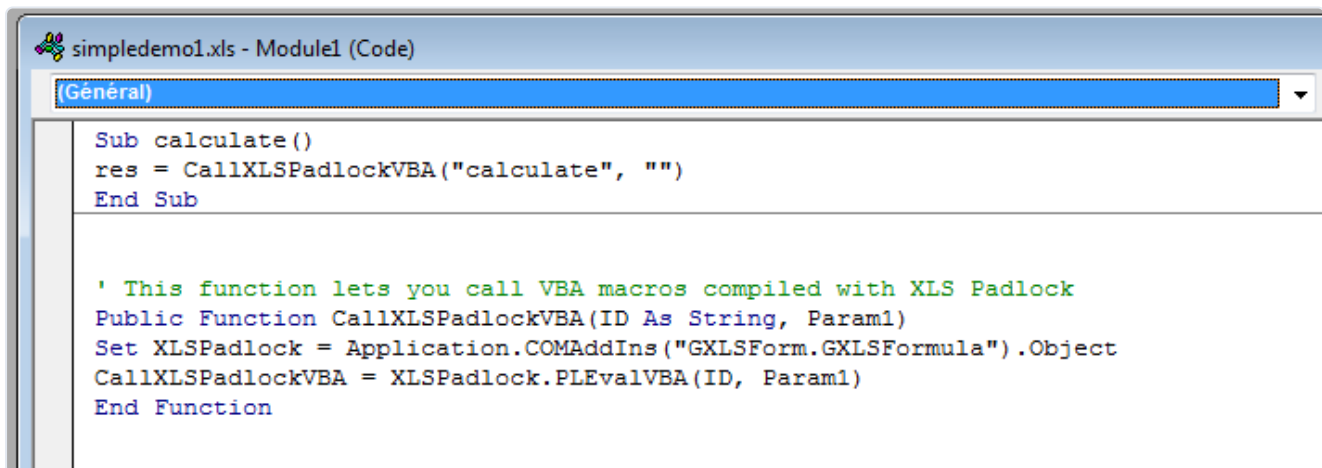
Pour un tutoriel sur la compilation d'une fonction VBA, vous pouvez [regarder cette vidéo](#).

Par exemple, considérez ce code d'origine :



```
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
```

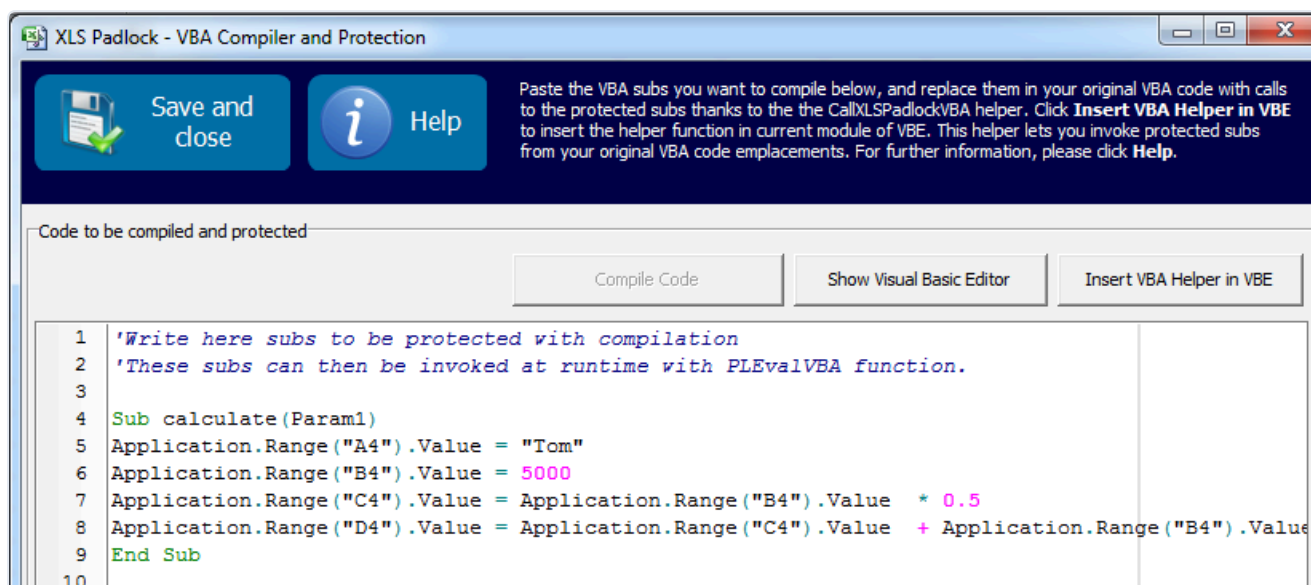
Une fois protégé, le sub `calculate()` d'origine est remplacé par un appel à une fonction interne de XLS Padlock qui exécute le bytecode compilé :



```
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
```

Le code d'origine est déplacé dans le XLS Padlock VBA Editor et compilé à partir de là :



Notez que certaines modifications peuvent être nécessaires, car le compilateur exige l'utilisation explicite de l'objet `Application` pour accéder aux objets Excel.

Le compilateur n'est pas un simple obfuscateur ; il transforme entièrement votre code VBA en binaire et le stocke de manière sécurisée dans l'application. Combiné à la protection par mot de passe de votre projet VBA, cela rend les outils de craquage de mot de passe inutiles, car ils ne peuvent pas accéder au fichier XLS d'origine.

Limitations du compilateur

- Il se peut que vous deviez modifier votre code VBA d'origine pour qu'il puisse être compilé, car le compilateur VBA de XLS Padlock n'est pas aussi avancé que l'interpréteur VB de Microsoft.
- Les macros entières ne peuvent généralement pas être compilées, seulement des portions de celles-ci.
- Certains objets et propriétés par défaut ne sont pas accessibles.

👉 Voir aussi : [Apprenez à écrire et compiler du code VBA sécurisé](#chapter-writing-and-compiling-secure-vba-code)

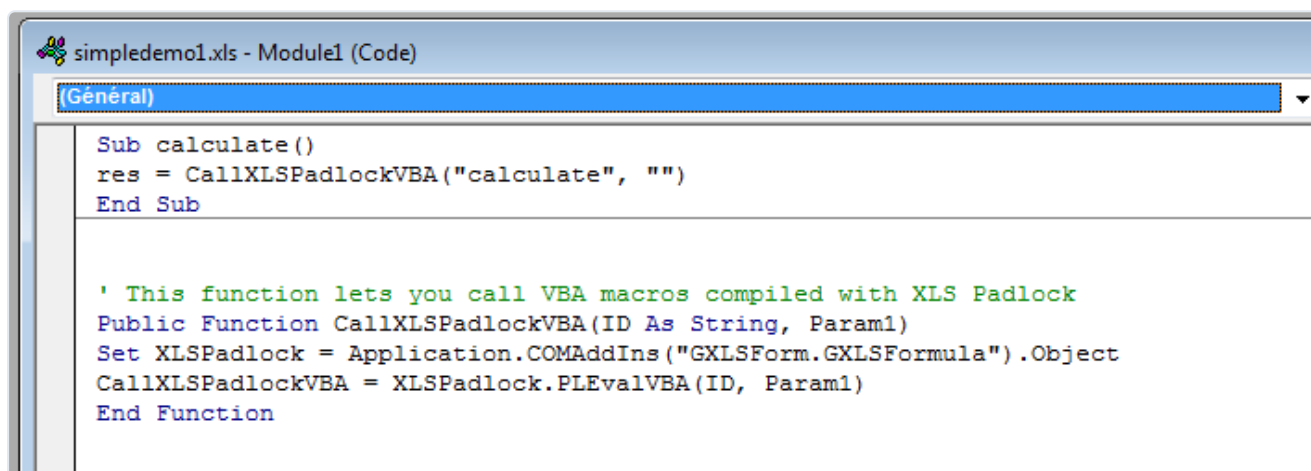
Véritable protection du code VBA

Si certains outils proposent l'obfuscation du VBA, ils rendent seulement votre code difficile à lire. Ils ne parviennent pas à dissimuler la logique fondamentale ni à empêcher sa copie.

👉 XLS Padlock va plus loin en vous permettant de **retirer entièrement le code VBA de votre classeur, tout en le conservant fonctionnel**. Cela est possible grâce au VBA compiler intégré.

Le **VBA compiler de XLS Padlock convertit votre code VBA en bytecode qui ne peut être exécuté qu'au sein de l'application sécurisée**. Si quelqu'un parvient à [accéder au fichier du classeur](#), il n'y trouvera pas le code original car celui-ci n'y est tout simplement pas. À la place, le **bytecode compilé est stocké de manière sécurisée à l'intérieur de l'EXE**.

Par exemple, la macro `calculate()` ci-dessous contenait à l'origine du code VBA propriétaire. Après compilation, la macro présente dans le classeur ne contient plus qu'un appel au code compilé. La logique d'origine a disparu.



```
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
```

Cela offre une **protection robuste** car le code VBA original n'existe plus dans le classeur. Il a été remplacé par du bytecode qui ne peut être ni copié ni utilisé dans un autre classeur Excel.

La contrepartie est que vous devez **transférer manuellement le code VBA que vous souhaitez protéger dans l'éditeur VBA de XLS Padlock**. La meilleure pratique consiste à identifier les parties les plus critiques de votre code, celles qui rendraient votre classeur non fonctionnel si elles manquaient, et à ne compiler que ces sections.

👉 Voir aussi : [En savoir plus sur l'utilisation de notre VBA compiler](#)

Référence de la syntaxe

Ce document fournit une référence de la syntaxe VBA prise en charge par le [VBA compiler de XLS Padlock](#).

Structure du script

La structure d'un script est constituée de déclarations de fonctions et de sous-routines (sub).

```
SUB DoSomething
    CallSomething
END SUB

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

Les instructions sur une même ligne peuvent être séparées par le caractère `:`.

Commentaires

Des commentaires peuvent être insérés dans un script. Vous pouvez utiliser le caractère `'` ou `REM`. Le commentaire s'étend jusqu'à la fin de la ligne.

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

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

Identifiants

Les noms d'identifiants (variables, fonctions, procédures, etc.) doivent commencer par un caractère (a-z, A-Z) ou un trait de soulignement `_`. Ils peuvent être suivis de caractères alphanumériques ou de traits de soulignement. Les noms ne peuvent contenir aucun autre caractère ni espace.

Variables

Il n'est pas nécessaire de déclarer les variables dans un script, mais si vous le souhaitez, vous pouvez déclarer une variable à l'aide de la directive `DIM`.

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

Vous pouvez également déclarer des variables globales en tant que `PRIVATE` ou `PUBLIC`.

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

Les variables déclarées avec `DIM` sont publiques par défaut. Les variables privées ne sont pas accessibles depuis d'autres scripts. Les variables peuvent également être initialisées lors de leur déclaration :

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

Instructions d'affectation

Les instructions d'affectation utilisent l'opérateur `=` pour affecter une valeur ou le résultat d'une expression à une variable ou à une propriété d'objet.

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

Tableaux

Le compilateur offre une prise en charge de base des constructeurs de tableaux et des tableaux variants. Pour construire un tableau, utilisez les caractères `[` et `]`. Vous pouvez construire un tableau à plusieurs indices en imbriquant des constructeurs de tableaux.

Dans le compilateur, les tableaux sont indexés à partir de 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"
```

Tableaux dynamiques :

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

Indices

Les chaînes de caractères, les tableaux et les propriétés de type tableau peuvent être indexés à l'aide des caractères [et]. Par exemple, si `Str` est une variable de type chaîne, l'expression `Str[3]` renvoie le troisième caractère de la chaîne.

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

Mots-clés et opérateurs

La syntaxe Basic prend en charge :

- **Déclarations** : `SUB...END SUB`, `FUNCTION...END FUNCTION`
- **Directives** : `BYREF`, `DIM`
- **Conditionnelles** : `IF...THEN...ELSE...ELSEIF...END IF`, `SELECT CASE...END SELECT`
- **Boucles** : `FOR...TO...STEP...NEXT`, `DO...WHILE...LOOP`, `DO...LOOP...WHILE`, `DO...UNTIL...LOOP`, `DO...LOOP...UNTIL`
- **Opérateurs** : `^`, `*`, `/`, `AND`, `+`, `-`, `OR`, `<`, `>=`, `<=`, `=`, `>`, `<`, `DIV`, `MOD`, `XOR`, `SHL`, `SHR`
- **Gestion des erreurs** : `TRY...EXCEPT`, `TRY...FINALLY`
- **Autres** : `EXIT`, constructeurs de tableaux `[1, 2, 3]`, accès aux objets `ObjectName.Property`

Instructions If

Il existe deux formes : `IF...THEN...END IF` et `IF...THEN...ELSE...END IF`. Si l'expression est vraie, les instructions `THEN` sont exécutées. Si elle est fausse, les instructions `ELSE` sont exécutées (si elles sont présentes).

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

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

Si l'instruction tient sur une seule ligne, vous n'avez pas besoin de `END IF` :

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

Instructions `Select Case`

Si `selectorExpression` correspond à l'une des expressions `caseexpr`, les instructions correspondantes seront exécutées. Sinon, l'instruction `CASE ELSE` sera exécutée.

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

Instructions `For`

L'instruction `FOR` répète l'exécution d'instructions jusqu'à ce qu'un compteur atteigne une valeur finale.

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

La partie `STEP` est facultative ; si elle est omise, la valeur du pas est de 1.

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

Instructions While

Une instruction `WHILE` répète des instructions tant qu'une condition de contrôle est vraie. La condition est évaluée avant l'exécution des instructions.

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

Instructions Do ... Loop

Les instructions sont exécutées `WHILE` (tant qu'une) expression est vraie, ou `UNTIL` (jusqu'à ce qu'une) expression soit vraie. La condition peut être testée avant ou après l'itération.

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

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

Déclarations de fonctions et de sous-routines

Les déclarations sont similaires au Basic standard. Pour renvoyer une valeur depuis une fonction, affectez-la à la variable implicite portant le même nom que la fonction, ou utilisez l'instruction `Return`. Les paramètres peuvent être passés par référence à l'aide de la directive `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
```

Les sous-routines (sub) et les fonctions sont publiques par défaut mais peuvent être déclarées comme `PRIVATE` .

Recettes de l'API VBA

Cette rubrique propose une collection complète d'extraits de code VBA (« recettes ») pour un contrôle avancé de votre application protégée à l'exécution. À l'aide de l'API VBA de XLS Padlock, vous pouvez récupérer des informations système, gérer l'enregistrement et le chargement, personnaliser l'interface utilisateur et interagir avec les fonctionnalités en ligne.

Les deux principaux objets de l'API sont : `Application.COMAddIns("GXLSForm.GXLSFormula").Object` pour la plupart des informations à l'exécution, et `Application.COMAddIns("GXLS.GXLSPLock").Object` pour les opérations liées à l'enregistrement.

Chaque recette ci-dessous fournit une fonction ou une sous-routine VBA prête à l'emploi.

Informations sur l'application et le système

Ces recettes vous aident à récupérer des informations sur l'environnement et l'état de l'application.

Vérifier si le classeur est protégé

Utilisez ceci pour confirmer que votre code s'exécute bien au sein d'une application protégée créée par 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
```

Obtenir le nom de fichier EXE de l'application

Récupère le nom du fichier `.exe` en cours d'exécution.

```

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

```

Obtenir la version du fichier EXE et du produit

Récupère les chaînes de version que vous avez définies dans la section « EXE Version Info » de 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

```

Utilisation : `ReadVersionInfo("ProductVersion")` renvoie la Product Version, et `ReadVersionInfo("FileVersion")` renvoie la File Version.

Obtenir le System ID de l'utilisateur

Récupère le System ID unique requis pour les Activation Keys verrouillées sur le matériel (hardware-locked). Il s'agit de l'identifiant que vos clients vous enverront.

```

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

```

Obtenir les paramètres de ligne de commande

Récupère les paramètres de ligne de commande passés au fichier `.exe` de votre application.

```
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
```

Utilisation : `ReadParamStr(1)` renvoie le premier paramètre, `ReadParamStr(2)` le deuxième, et ainsi de suite.

Enregistrement et chargement des classeurs

Gérez les fichiers d'enregistrement sécurisés (`.xlsc` ou `.xlsce`) par programmation.

Obtenir le chemin du fichier d'enregistrement actuel

Récupère le chemin complet du fichier d'enregistrement sécurisé actuellement chargé.

```
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
```

Obtenir le chemin du dossier d'enregistrement local

Récupère le chemin du dossier local où XLS Padlock stocke les paramètres de votre application et les fichiers d'enregistrement sécurisés.

```
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
```

Exécuter du code VBA après l'enregistrement

XLS Padlock peut appeler automatiquement la sous-routine `XLSPadLock_OnAfterSave` après qu'un utilisateur a enregistré son travail. Ceci est utile pour la journalisation, l'affichage d'un message de confirmation ou d'autres actions postérieures à l'enregistrement.

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

Ouvrir un fichier d'enregistrement existant avec VBA

Ouvrez un fichier d'enregistrement sécurisé par programmation. Notez que cela chargera le fichier dans une **nouvelle instance d'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
```

Suggérer un nom de fichier pour la boîte de dialogue d'enregistrement

Définit le nom de fichier par défaut qui apparaît dans la boîte de dialogue « 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.xlsc"
' Call SetSecureWorkbookFilename("D:\My Documents\my save.xlsc")
```

Enregistrer une copie sécurisée sans invite

Enregistrez une copie sécurisée du classeur directement vers un chemin de fichier spécifié, sans afficher la boîte de dialogue « 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.xlsc") Then
'     MsgBox "Workbook saved successfully."
' End If
```

Migrer les données utilisateur depuis une version précédente

Si vous publiez une nouvelle version de votre classeur, vous pourriez avoir besoin d'importer des données depuis un fichier d'enregistrement créé avec une version précédente.

→ [Découvrez comment migrer les données utilisateur avec VBA](#)

Interface utilisateur

Personnalisez les éléments d'interface tels que les boîtes de dialogue.

Masquer par programmation les boîtes de dialogue d'attente et de chargement

Vous pouvez masquer la boîte de dialogue « Loading workbook, please wait... » à l'aide d'un appel VBA plutôt que d'attendre sa fermeture automatique.

```
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
```

Activation et validation en ligne

Interagissez avec les fonctionnalités en ligne de XLS Padlock.

Vérifier la présence d'une connexion Internet

Teste si une connexion Internet active est disponible sur l'ordinateur de l'utilisateur.

```
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
```

Obtenir le jeton d'activation en ligne

Récupère un hachage du jeton d'activation renvoyé par le serveur web après une activation en ligne réussie.

```
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
```

Vérifier si la validation en ligne a réussi

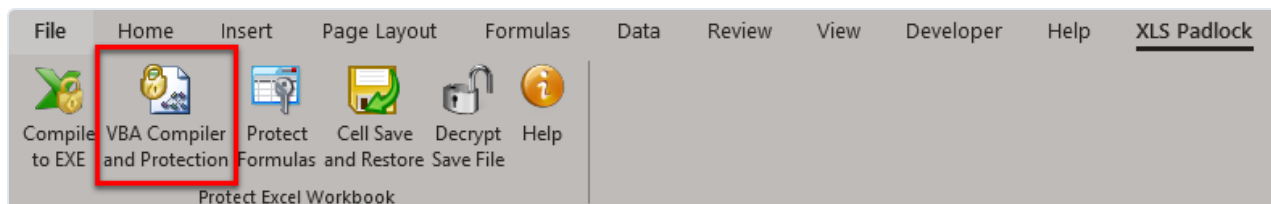
Obtient le résultat de la dernière tentative de validation en ligne.

```
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
```

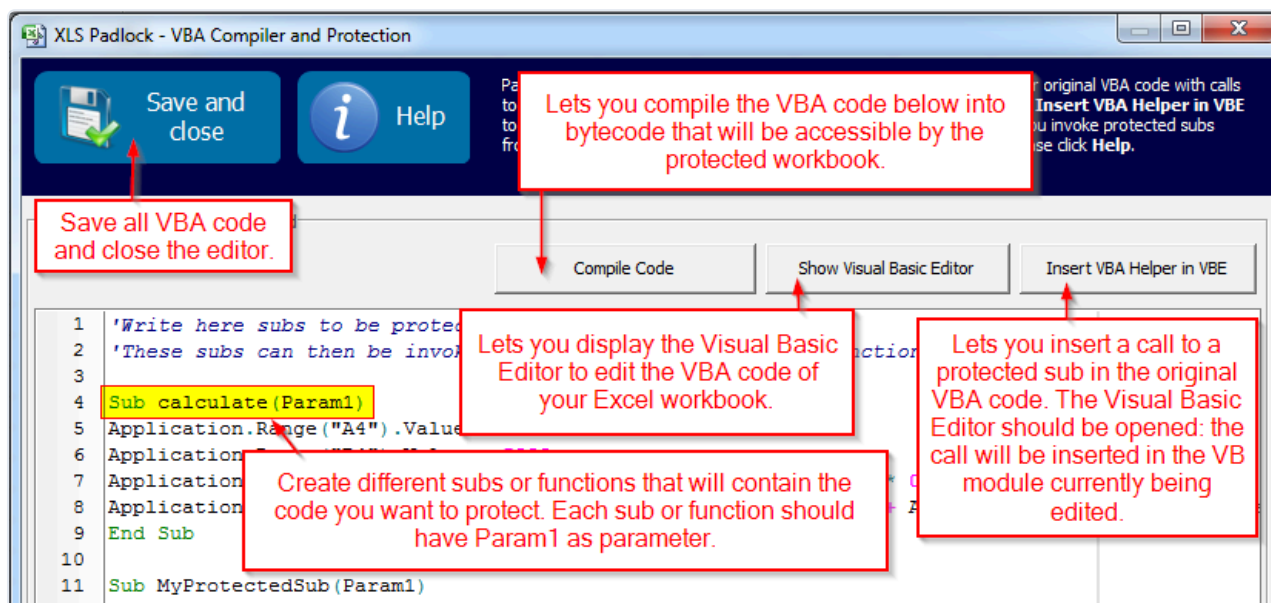
Écrire et compiler du VBA sécurisé

Vous pouvez **protéger les parties sensibles de votre code VBA en les compilant en bytecode avec XLS Padlock**. Nous recommandons de combiner le bytecode compilé avec votre code VBA existant pour une approche équilibrée entre sécurité et développement.

1. Dans Excel, cliquez sur « VBA Compiler and Protection » pour ouvrir l'éditeur et le compilateur VBA de XLS Padlock :



L'éditeur VBA de XLS Padlock s'ouvre :



2. Dans la zone de texte, saisissez le code VBA que vous souhaitez compiler. Vous pouvez regrouper votre code en `subs` et `fonctions`. Pour simplifier ce processus, vous pouvez ouvrir le Visual Basic Editor en cliquant sur « Show Visual Basic Editor » et copier/coller votre code.

Par défaut, tous les subs et fonctions ont un paramètre nommé `Param1`. Vous pouvez toutefois utiliser davantage de paramètres.

3. Lorsque vous avez terminé, cliquez sur « **Compile Code** ». Le code VBA est instantanément compilé en bytecode, que XLS Padlock conservera jusqu'à ce que vous compiliez le fichier EXE de votre application.

CONSEIL

Si une erreur est détectée, elle s'affiche et le code correspondant est souligné en rouge.

1. Cliquez sur « Save and close » pour fermer l'éditeur.

👉 Après avoir compilé votre code, vous devrez **modifier vos macros VBA originales pour appeler la version compilée**. Découvrez comment procéder dans le guide [Invocation du code VBA compilé à l'exécution](#).

Invoquer le VBA compilé à l'exécution

Pour exécuter votre [code VBA compilé](#), vous devez l'appeler depuis le code VBA ordinaire de votre classeur à l'aide d'une fonction d'assistance.

Tout d'abord, ouvrez le Visual Basic Editor et collez la fonction d'assistance suivante dans un module de votre classeur :

```
' 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
```

Insertion automatique

Cette fonction peut être insérée automatiquement dans votre module VB actif en cliquant sur le bouton « Insert VBA Helper in VBE » dans l'éditeur VBA de XLS Padlock.

La fonction d'assistance `CallXLSPadlockVBA` prend deux paramètres :

- **ID** : le nom du `Sub` ou de la `Function` compilé que vous souhaitez invoquer.
- **Param1** : un paramètre optionnel que vous souhaitez transmettre au code compilé.

Exemple

Imaginez que votre module VBA original contienne le code suivant :

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

Après avoir déplacé ce code dans le VBA Compiler, vous remplacerez le sub original par un appel à la fonction d'assistance :

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

Ici, nous fournissons le nom du sub protégé, `Calculate`, et passons une chaîne vide pour le second paramètre, car il n'est pas utilisé dans ce cas. La variable `res` contiendra le résultat de l'appel, ce qui est surtout utile si le code invoqué est une `Function` qui renvoie une valeur.

Exécution au moment de la conception

L'une des fonctionnalités puissantes de XLS Padlock est que cette configuration fonctionne même au moment de la conception. Si vous exécutez le sub `Calculate()` depuis le VBE avant de compiler le classeur, XLS Padlock exécutera le code protégé, ce qui vous permet de tester votre logique sans avoir à compiler l'application au préalable.

👉 Voir aussi : découvrez comment [transmettre plusieurs paramètres à votre code compilé](#chapter-passing-more-parameters-to-the-compiled-vba-code).

Accéder aux objets Excel

Lorsque vous écrivez du code pour le compilateur VBA, vous devez utiliser explicitement l'objet `Application` pour accéder au modèle objet d'Excel.

Pour une référence détaillée des propriétés et méthodes disponibles, consultez la [documentation du Microsoft Office Dev Center](#).

Exemples

Référencer un Range

Pour référencer une cellule ou une plage de cellules de votre feuille de calcul, vous devez préfixer l'objet `Range` avec `Application`.

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

Utiliser les fonctions de feuille de calcul

Pour utiliser les fonctions de feuille de calcul d'Excel, vous devez utiliser l'objet `Application.WorksheetFunction`.

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

Les fonctions exigent tous les paramètres

Lorsque vous appelez des fonctions Excel intégrées comme `InputBox` depuis du code compilé, vous **devez spécifier tous les paramètres**, car les paramètres par défaut ne sont pas pris en charge.

Exemple complet : conversion d'une macro

Voici une 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

```

Voici le même code modifié pour le compilateur VBA de XLS Padlock. Notez l'ajout de l'objet `Application` et des paramètres explicites pour `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

```

Votre module VBA original serait ensuite modifié pour appeler la version compilée :

```

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

```

Passer des tableaux

Vous pouvez passer différents types de variables au code VBA compilé, y compris des tableaux statiques. Supposons que vous ayez la fonction suivante dans le VBA Compiler :

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

Dans votre module VBA Excel normal, vous pouvez appeler cette fonction et passer un tableau.

Avertissement

Le tableau doit être défini comme un **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
```

Passer davantage de paramètres

Par défaut, la fonction d'assistance utilisée pour exécuter le code VBA compilé ne prend en charge qu'un seul paramètre.

```
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
```

La méthode `PLEvalVBA` de l'objet `XLSPadlock` prend deux paramètres : l'**ID** de la sub/fonction compilée et un paramètre unique **Param1**.

Pour passer plus de paramètres, l'objet `XLSPadlock` fournit des méthodes supplémentaires :

- `XLSPadlock.PLEvalVBA2(ID, Param1, Param2)` pour deux paramètres.
- `XLSPadlock.PLEvalVBA3(ID, Param1, Param2, Param3)` pour trois paramètres.

Il est également possible de [passer plus de paramètres en utilisant des tableaux](#).

Par exemple, une fonction d'assistance pour passer deux paramètres ressemblerait à ceci :

```
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
```

Gérer les erreurs

Des erreurs d'exécution peuvent se produire lorsque votre code VBA compilé est exécuté. Alors que le VBA standard utilise l'instruction `On Error` pour la gestion des erreurs, le VBA Compiler de XLS Padlock utilise un bloc `Try...Except`.

Lorsqu'une erreur se produit dans le bloc `Try` (ou dans toute procédure qu'il appelle), le compilateur passe immédiatement au bloc `Except` pour la gérer.

Syntaxe

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

CONSEIL

Une erreur non gérée sera affichée par le VBA Compiler à l'exécution, [sauf si vous désactivez cette option](#).

Exemple de code

```
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
```

Erreur OLE 800A03EC

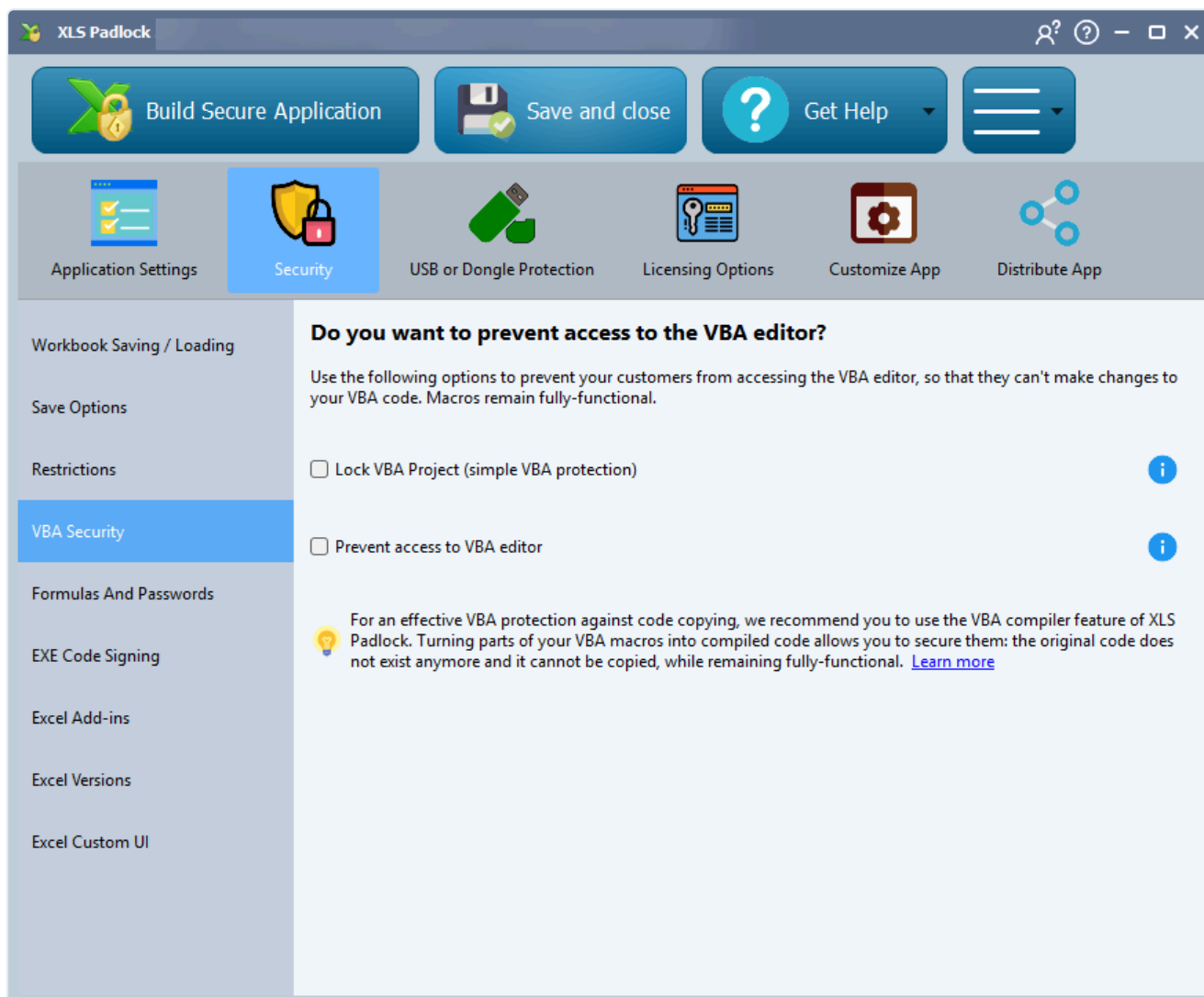
Pour résoudre cette erreur, assurez-vous que l'option « **Trust access to the VBA project object model** » est activée dans les paramètres du Centre de gestion de la confidentialité d'Excel.

Pour savoir comment trouver cette option, veuillez consulter la page d'assistance officielle de Microsoft Office :

[Activer ou désactiver les macros dans les fichiers Office](#)

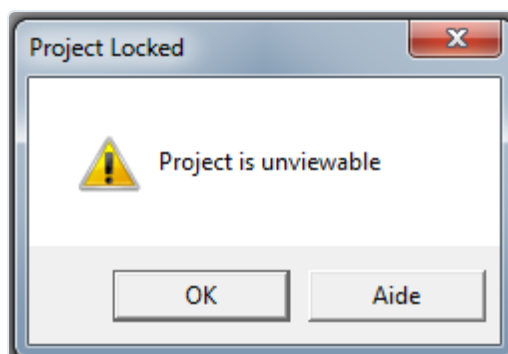
Protection du code VBA

Votre code VBA est un actif précieux. XLS Padlock offre plusieurs niveaux de protection pour le sécuriser contre toute altération ou tout vol. Vous trouverez ci-dessous deux options de sécurité fondamentales, suivies de notre solution la plus avancée pour une protection maximale.



Option 1 : verrouiller votre projet VBA

L'étape la plus simple consiste à [verrouiller votre projet VBA](#). Cela rend votre code non consultable dans l'éditeur Visual Basic. Si un utilisateur tente d'accéder au projet, il rencontrera un message d'erreur, ce qui bloque efficacement l'inspection occasionnelle de votre code.



Option 2 : empêcher l'accès à l'éditeur VBA

Pour un niveau de sécurité supplémentaire, vous pouvez [empêcher complètement l'ouverture de l'éditeur VBA](#). XLS Padlock surveille en permanence et ferme automatiquement toute tentative d'ouverture du VBE, ce qui garantit que même les utilisateurs avertis sont bloqués.

Protection ultime : la compilation VBA

Pour le plus haut niveau de sécurité, XLS Padlock vous permet de **compiler votre code VBA en code binaire natif**. Contrairement à la simple obfuscation, qui rend uniquement le code difficile à lire, la compilation transforme votre logique dans un format non lisible par un humain, qui est presque impossible à rétro-concevoir. Votre code VBA d'origine est retiré du classeur et remplacé par des appels au code compilé sécurisé contenu dans l'EXE. C'est la défense ultime pour votre propriété intellectuelle.

→ [En savoir plus sur notre puissant VBA Compiler](#)

Masquer et verrouiller le code VBA

XLS Padlock propose une fonctionnalité **Lock VBA Project** dédiée. Cependant, si vous choisissez de ne pas l'utiliser, vous pouvez toujours recourir à la protection native d'Excel. XLS Padlock rend cette méthode plus sécurisée.

Comment verrouiller votre projet VBA dans Excel

1. Dans Excel, ouvrez l'éditeur Visual Basic (VBE) en appuyant sur **Alt+F11** ou en le sélectionnant depuis l'onglet **DeveLoper**.
2. Dans le VBE, allez dans le menu **Tools** et sélectionnez **VBAProject Properties....**
3. Dans la boîte de dialogue, allez dans l'onglet **Protection**.
4. Cochez la case **Lock project for viewing** et saisissez un mot de passe fort.

Une fois le classeur enregistré et rouvert, les utilisateurs ne pourront ni consulter ni modifier le code VBA sans le mot de passe.

Pourquoi cette méthode est plus sécurisée avec XLS Padlock

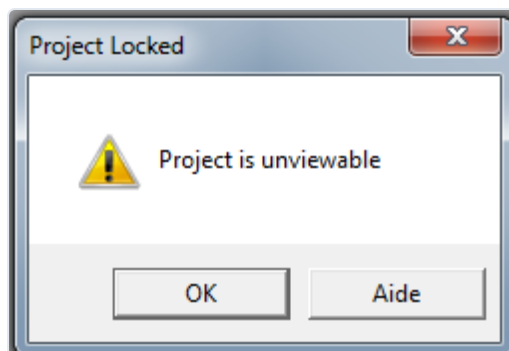
Normalement, la protection par mot de passe VBA d'Excel est considérée comme faible, car de nombreux outils existent pour supprimer le mot de passe. Ces outils fonctionnent en accédant directement au fichier Excel.

Cependant, lorsque votre classeur est compilé en un EXE avec XLS Padlock, le fichier **.xls** ou **.xlsm** d'origine est chiffré et intégré à l'application. Les outils de cassage de mot de passe ne peuvent pas accéder à ce fichier sécurisé, ce qui les rend inutiles. Cette combinaison offre une défense bien plus solide pour votre code VBA.

Verrouiller le projet VBA

L'option **Lock VBA Project** offre un moyen simple d'empêcher les utilisateurs finaux d'accéder à votre projet VBA. Cette fonctionnalité marque le projet VBA comme verrouillé, ce qui rend impossible son affichage, son accès ou sa modification. Elle n'utilise pas de protection par mot de passe.

Si l'utilisateur final tente d'accéder à un projet VBA verrouillé, le message d'erreur suivant s'affiche (Project Locked, Project is unviewable) :



Supprimer les mots de passe VBA existants

Si vous utilisez cette option, nous vous recommandons de supprimer tout mot de passe existant de votre projet VBA. Un mot de passe devient redondant lorsque le projet est verrouillé. De plus, l'utilisation d'un mot de passe long avec cette option peut provoquer des plantages dans Excel 2007.

Si vous préférez conserver un mot de passe, envisagez plutôt d'utiliser l'option [Empêcher l'accès à l'éditeur VBA](#chapter-forbid-access-to-the-vba-editor-vbe).

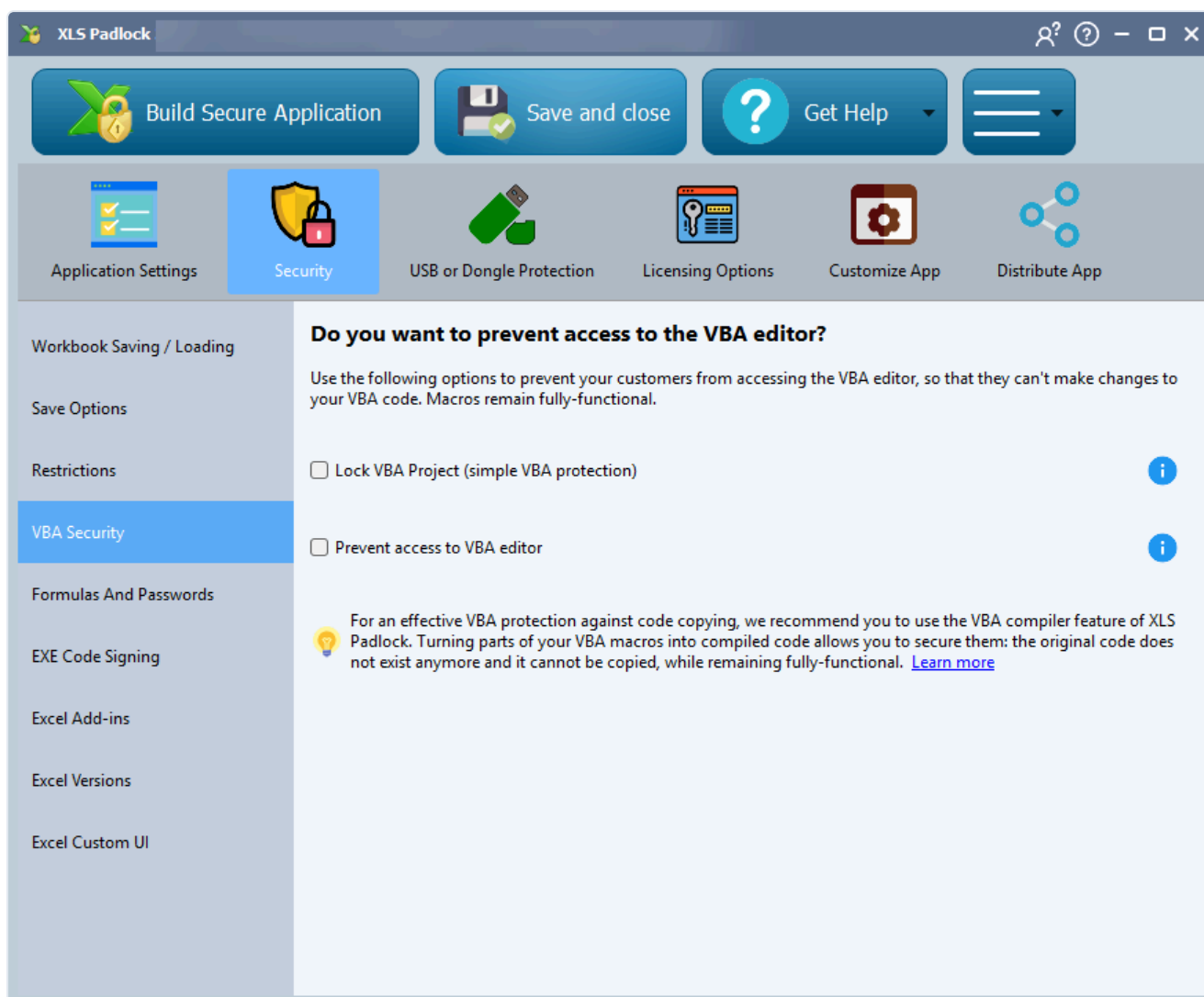
Notez que les macros restent fonctionnelles.

Cette option est compatible avec notre [compilateur VBA](#), qui offre un niveau de sécurité plus élevé pour votre code VBA.

Interdire l'accès au VBE

XLS Padlock vous propose des options pour **empêcher vos clients d'accéder à l'éditeur VBA (VBE)**, les empêchant ainsi de visualiser ou de modifier votre code VBA tout en garantissant que vos macros restent pleinement fonctionnelles.

Ces options sont disponibles dans l'onglet "Security" de XLS Padlock.



Lock VBA Project (protection VBA simple)

Cette option verrouille votre projet VBA de manière programmatique afin d'en bloquer l'accès. Bien qu'elle offre un niveau de protection basique, elle peut être contournée par certains outils.

Prevent Access to VBA Editor

Il s'agit d'une mesure de sécurité plus forte. XLS Padlock effectue une vérification régulière, et si un utilisateur tente d'ouvrir le Visual Basic Editor (VBE), XLS Padlock ferme immédiatement la fenêtre du VBE. Cela les empêche même de tenter de visualiser le code.

Cette option est compatible avec l'option "Lock VBA Project" et notre puissant [compilateur VBA](#).

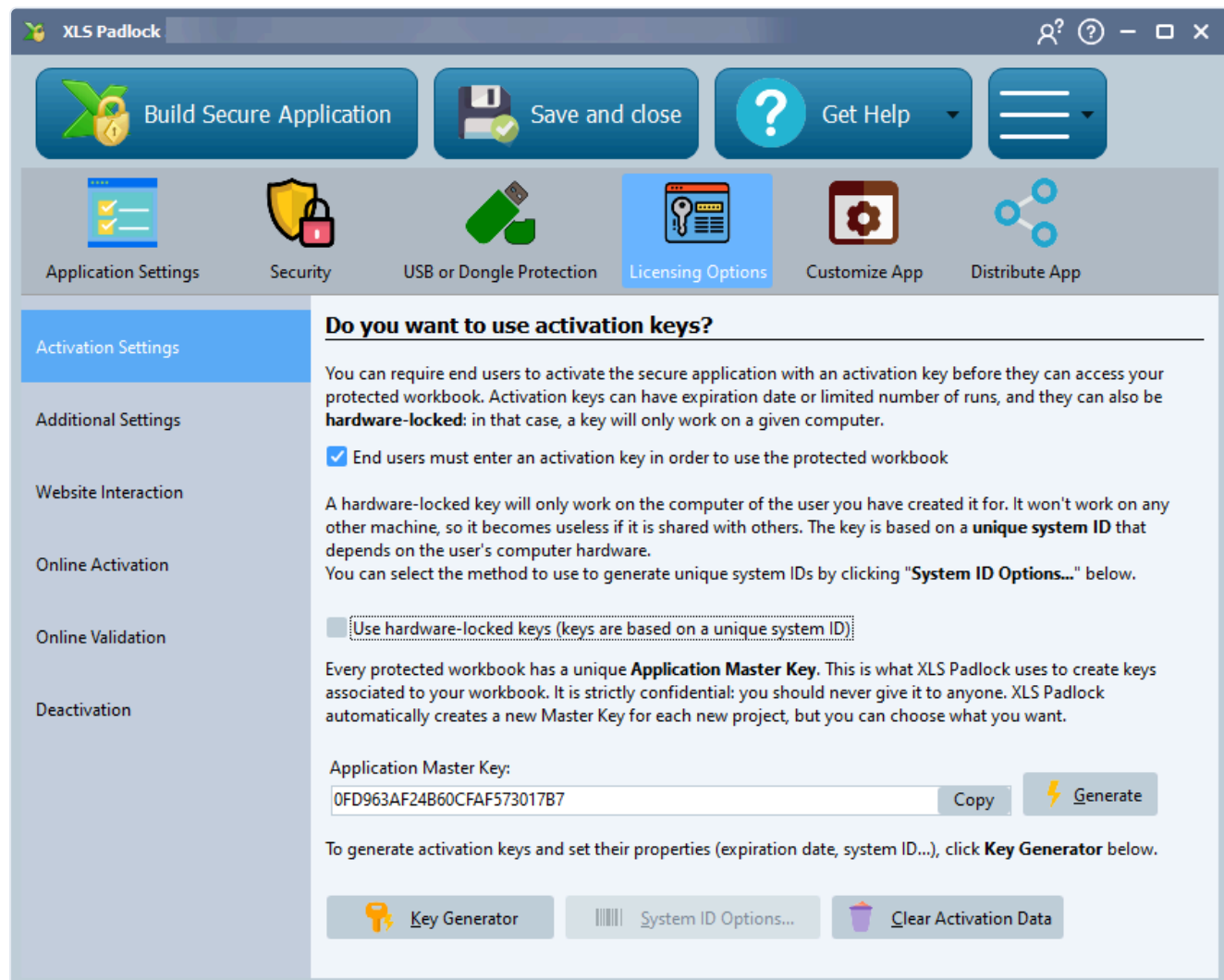
Pour une protection ultime du code VBA

👉 Pour une protection robuste contre le vol de code et la rétro-ingénierie, nous recommandons vivement d'utiliser la [fonctionnalité de compilateur VBA](#).

En compilant des parties de vos macros VBA, le code source d'origine est supprimé et remplacé par du code binaire. Ce code compilé est extrêmement difficile à rétro-concevoir, ce qui garantit la sécurité de vos algorithmes et de votre logique métier. C'est le moyen le plus efficace de protéger votre propriété intellectuelle VBA.

Guide d'activation et de licence

Ce guide couvre les différentes options liées aux **clés d'activation**, à la **licence**, à la **validation en ligne** et à la **désactivation** pour vos applications protégées.



Concepts fondamentaux

Activer les clés d'activation

Pour exiger des utilisateurs qu'ils disposent d'une clé pour utiliser votre application, activez l'option : « **End users must enter an activation key in order to use the protected workbook** » (les utilisateurs finaux doivent saisir une clé d'activation pour utiliser le classeur protégé). C'est l'interrupteur principal pour activer les fonctionnalités de licence.

Application Master Key

Chaque classeur protégé possède une **Application Master Key** unique. XLS Padlock utilise cette clé pour générer les clés d'activation associées à votre classeur. Elle est **strictement confidentielle** et ne doit jamais être partagée. XLS Padlock crée automatiquement une nouvelle Master Key pour chaque projet.

Application GUID et Secret Key

- L'**Application GUID** est un identifiant unique utilisé par l'application pour stocker ses paramètres et gérer les fichiers de sauvegarde sur l'ordinateur d'un utilisateur.
- L'**Application Secret Key** est utilisée pour chiffrer et sécuriser les fichiers de sauvegarde de l'utilisateur (`.XLSC` ou `.XLSCE`). Elle garantit que les fichiers de sauvegarde créés avec votre application ne peuvent être ouverts que par votre application.

IMPORTANT

Si vous modifiez l'Application GUID ou la Secret Key, les fichiers de sauvegarde précédents ne seront plus accessibles par vos utilisateurs.

Publier une mise à jour qui doit invalider les anciennes sauvegardes utilisateur

Pour forcer une nouvelle version à ignorer tout fichier de sauvegarde créé avec les versions précédentes, par exemple lorsque la structure du classeur a suffisamment changé pour que les anciennes sauvegardes ne doivent pas être rechargées, générez une **nouvelle Application Secret Key** avant de recompiler, mais **laissez l'Application GUID inchangé** afin que les clés d'activation existantes restent valides. Au premier lancement par l'utilisateur, l'ancien fichier de sauvegarde sera détecté comme incompatible et il sera invité à charger à la place le classeur d'origine intégré. Consultez [Mises à jour du classeur](#) pour le flux de travail complet de mise à jour.

CONSEIL

Le statut d'activation (essai / enregistré) peut être récupéré via du code VBA.

Permettre à l'utilisateur final de changer la clé d'activation

Un utilisateur peut changer sa clé d'activation en exécutant l'application avec le [commutateur en ligne de commande](#) `-enterkey` (par exemple, `MYAPP.EXE -enterkey`). Cela est utile pour remplacer une clé qui est sur le point d'expirer.

Effacer les données d'activation

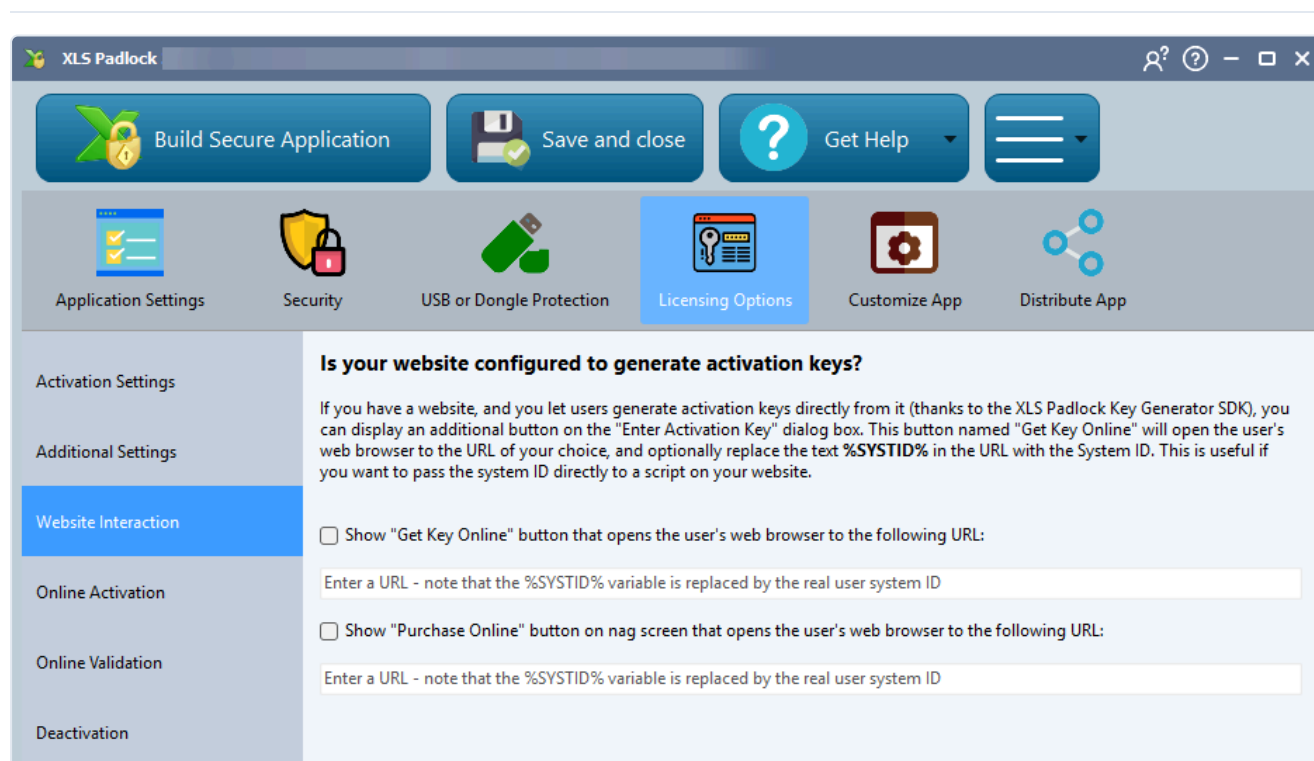
À des fins de test sur votre propre machine, le bouton « **Clear Activation Data** » dans le Key Generator réinitialise toutes les informations d'activation stockées localement.

Clés verrouillées au matériel

Une clé verrouillée au matériel ne fonctionnera **que sur l'ordinateur spécifique** pour lequel elle a été générée, ce qui la rend inutile si elle est partagée. La clé est basée sur un **System ID unique** dérivé du matériel de l'ordinateur de l'utilisateur.

- **Options du System ID** : vous pouvez sélectionner les composants matériels (CPU, adresse MAC, etc.) utilisés pour générer le System ID. Soyez cohérent avec ces options après le déploiement de votre application.
- **Obtenir le System ID** : lorsqu'un utilisateur exécute l'application, son System ID lui est affiché, qu'il doit vous envoyer. Vous utilisez ensuite cet ID dans le Key Generator pour créer sa clé.
- **Automatisation en ligne** : pour éviter l'échange manuel des System ID, vous pouvez utiliser les fonctionnalités d'[activation en ligne](#).

Interaction avec le site web



Bouton Get Key Online

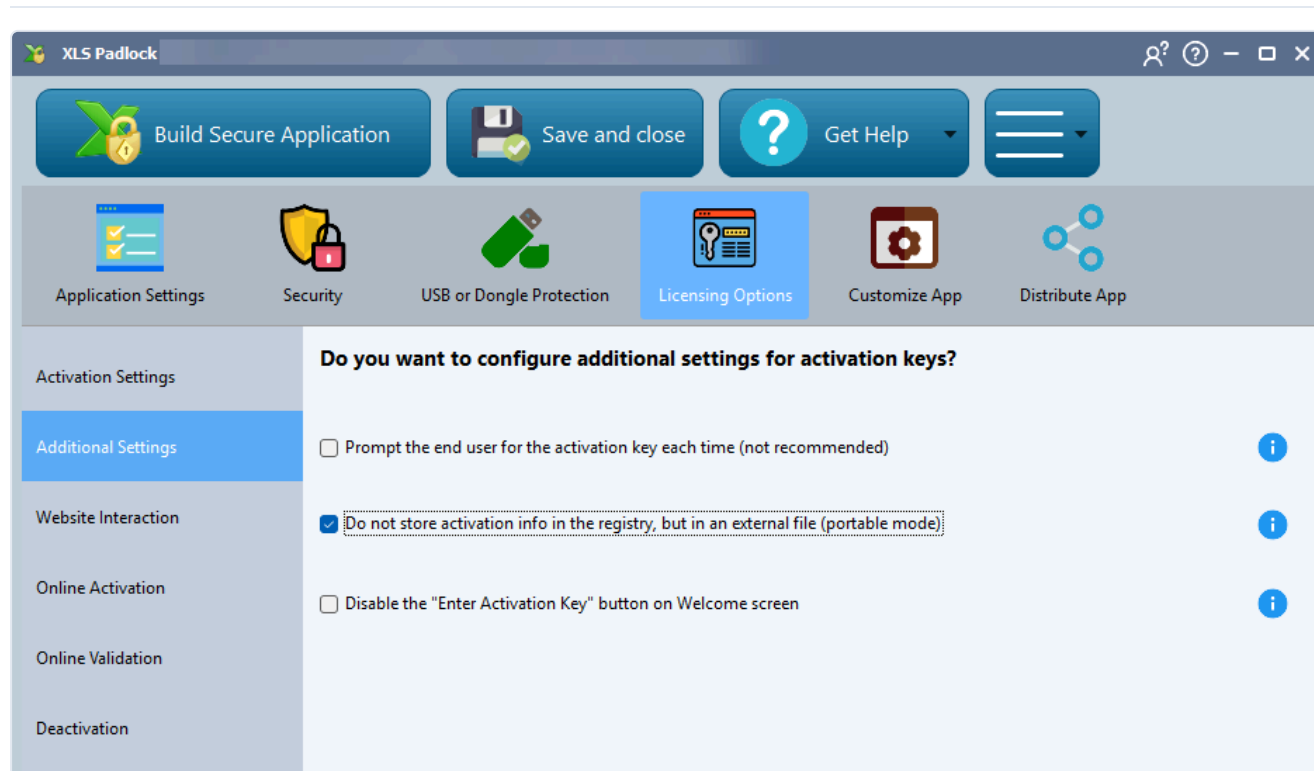
Vous pouvez afficher un bouton « Get Key Online » sur la fenêtre d'activation. Ce bouton ouvre le navigateur web de l'utilisateur à une URL que vous spécifiez. Vous pouvez utiliser l'espace réservé `%SYSTID%` dans l'URL pour transmettre automatiquement le System ID de l'utilisateur au script de génération de clé de votre site web.

Exemple d'URL : `https://www.yourwebsite.com/getkey.php?systid=%SYSTID%`

Afficher le bouton « Purchase Online »

Pour les versions d'essai avec un écran de rappel (nag screen), vous pouvez ajouter un bouton « Purchase Online » qui dirige l'utilisateur vers votre boutique ou votre page d'achat.

Paramètres supplémentaires



Mode portable

Activez l'option « **Do not store activation info in the registry, but in an external file (portable mode)** » (ne pas stocker les informations d'activation dans le registre, mais dans un fichier externe, mode portable). Cela crée un fichier `.lic` masqué dans le même dossier que votre EXE, ce qui permet d'exécuter l'application depuis une clé USB sans écrire dans le registre Windows.

IMPORTANT

Ce mode n'est pas recommandé pour les versions d'essai, car la suppression du fichier `.lic` réinitialise la période d'essai.

Demander la clé d'activation à l'utilisateur final à chaque fois

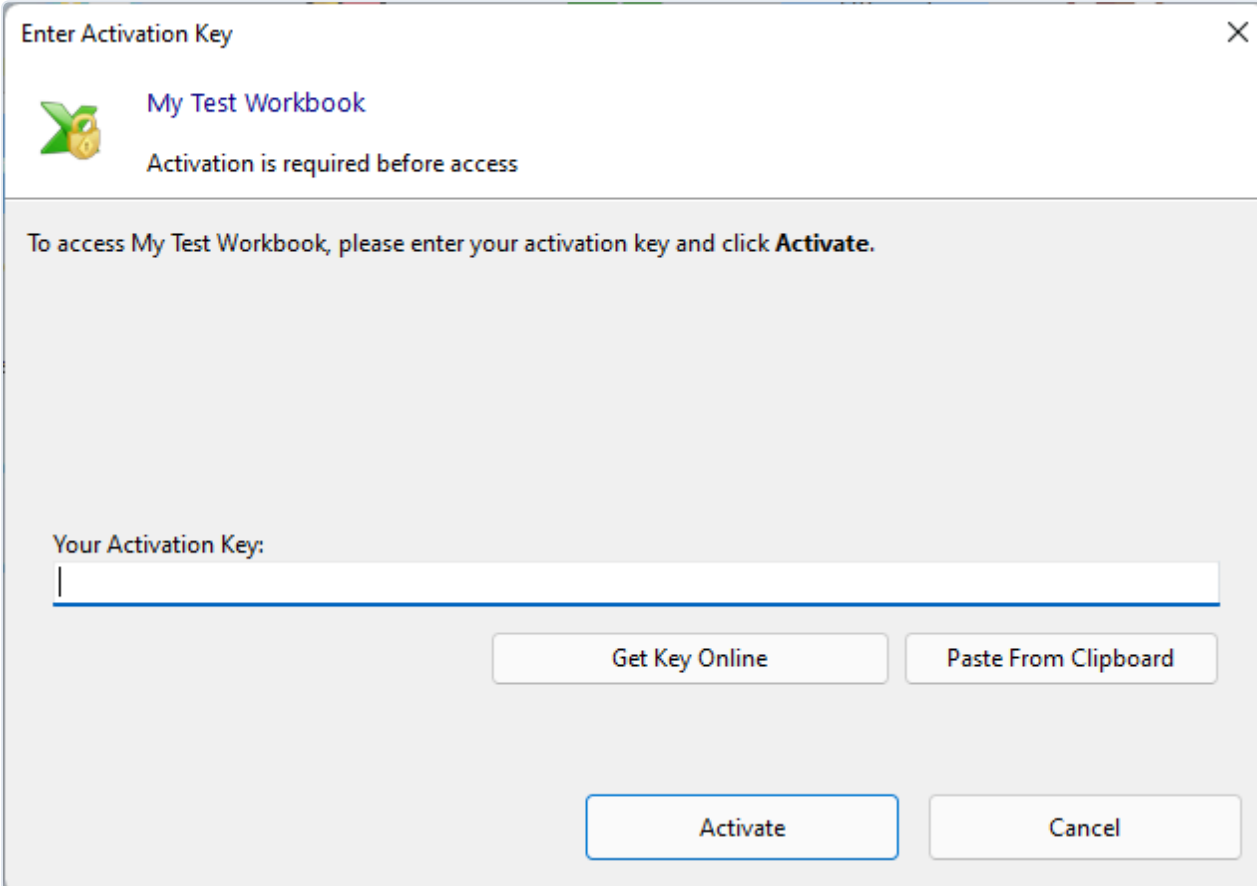
En activant « Prompt the end user for the activation key each time » (demander la clé d'activation à l'utilisateur final à chaque fois), vos utilisateurs devront saisir la clé d'activation à chaque ouverture de votre application. Ce n'est pas recommandé, car c'est fastidieux pour votre client, et vous ne pouvez pas définir de fonctionnalités d'expiration pour la clé.

Désactiver le bouton « Enter Activation Key » sur l'écran d'accueil

Cela supprimera la possibilité pour vos utilisateurs finaux de saisir une clé d'activation lorsqu'ils ouvrent votre application de classeur Excel. Il n'est toutefois pas recommandé d'utiliser cette option.

Clés d'activation

Les clés d'activation sont une fonctionnalité centrale de XLS Padlock, offrant un **puissant système de licence pour vos classeurs Excel**. Ce système vous permet d'exiger des utilisateurs finaux qu'ils activent l'application avec une clé avant de pouvoir accéder à votre classeur protégé :



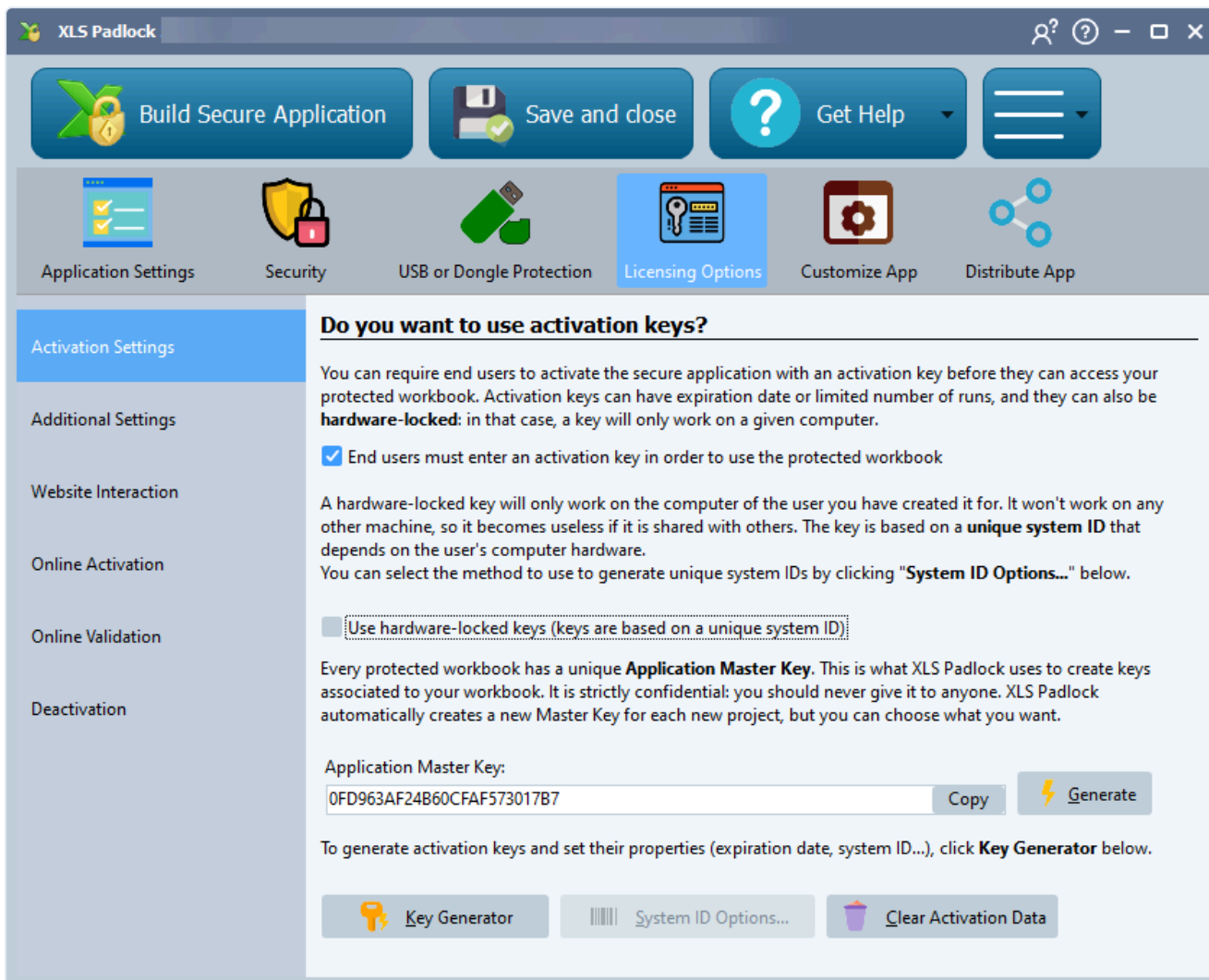
The screenshot shows a dialog box titled "Enter Activation Key" with a close button (X) in the top right corner. On the left, there is a green Excel icon with a gold padlock. To its right, the text reads "My Test Workbook" and "Activation is required before access". Below this, a message states: "To access My Test Workbook, please enter your activation key and click **Activate**." There is a text input field labeled "Your Activation Key:" with a cursor inside. Below the input field are four buttons: "Get Key Online", "Paste From Clipboard", "Activate", and "Cancel".

👉 Les clés d'activation peuvent être configurées avec une [date d'expiration](#) ou un [nombre limité d'exécutions](#). Elles peuvent également être [verrouillées au matériel](#), ce qui signifie qu'une clé ne fonctionnera que sur un ordinateur spécifique.

CONSEIL

En plus du [Key Generator](#) intégré, nous proposons un générateur de clés autonome capable de créer des milliers de clés à la fois, ainsi qu'un SDK de générateur de clés pour l'intégration avec votre serveur ou votre site web.

Enfin, vous pouvez mettre en œuvre des fonctionnalités d'[activation en ligne](#) et de [validation](#) pour contrôler à distance la façon dont les utilisateurs accèdent à votre classeur et automatiser la diffusion des clés d'activation.



👉 Pour commencer, veuillez consulter notre tutoriel sur [comment configurer les clés d'activation](#).

Configurer les clés d'activation

Les **clés d'activation** constituent un puissant système de licence qui vous permet de contrôler qui peut accéder à vos classeurs protégés. Ce guide vous montre comment démarrer.

1. Activer les clés d'activation

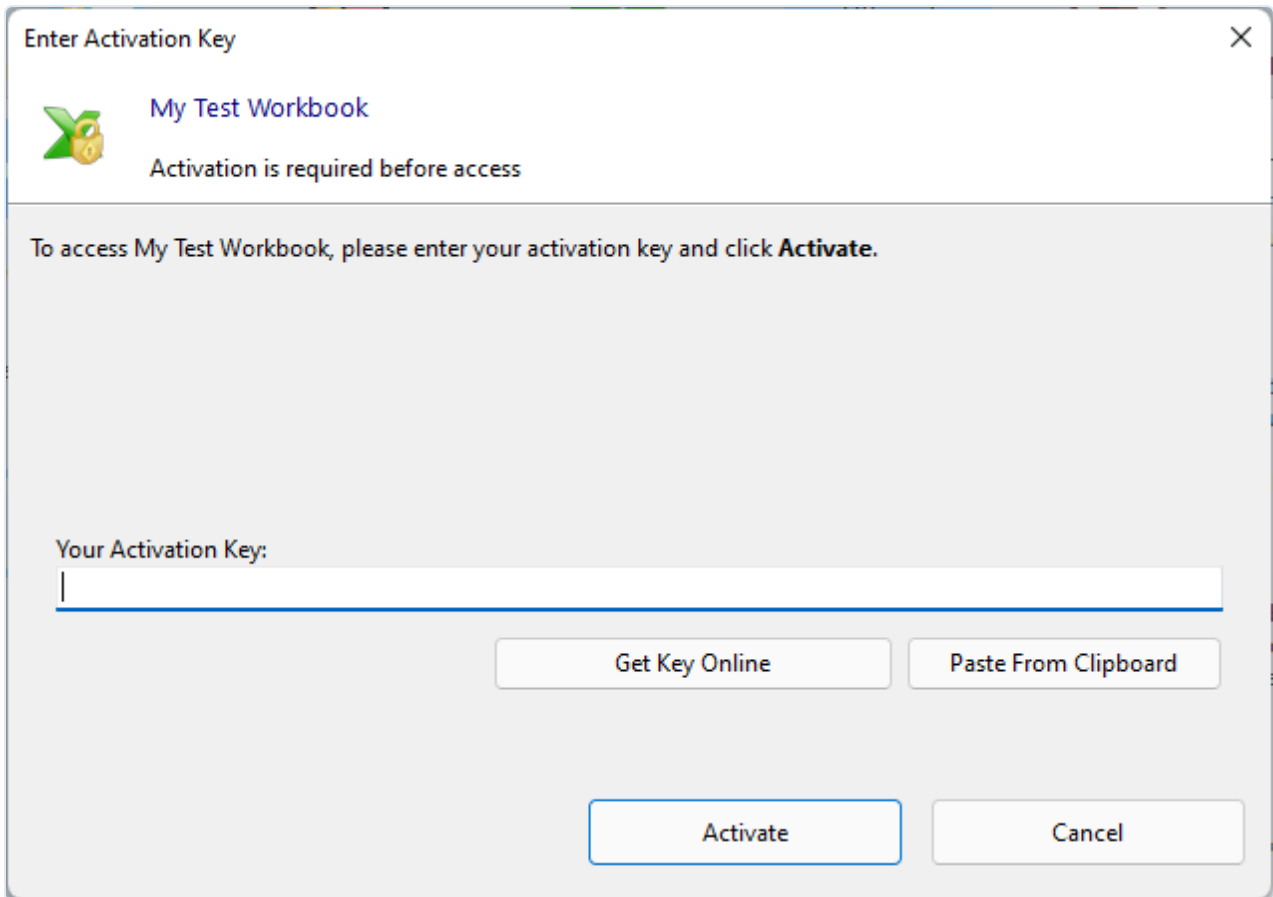
Dans les paramètres de sécurité de XLS Padlock, cochez la case **"End users must enter an activation key in order to use the protected workbook"** (les utilisateurs finaux doivent saisir une clé d'activation pour utiliser le classeur protégé).

The screenshot shows the XLS Padlock application settings window. The 'Licensing Options' tab is active, displaying the 'Do you want to use activation keys?' section. A red box highlights the checked checkbox: **End users must enter an activation key in order to use the protected workbook**. Below this, there is a checkbox for **Use hardware-locked keys (keys are based on a unique system ID)**. The 'Application Master Key' field shows the value `0FD963AF24B60CFAF573017B7` with a 'Copy' button and a 'Generate' button. At the bottom, there are buttons for 'Key Generator', 'System ID Options...', and 'Clear Activation Data'.

Une fois cela fait, reconstruisez votre application.

2. Tester votre application

Lorsque vous exécutez l'application nouvellement compilée, vous serez désormais invité à saisir une clé d'activation avant de pouvoir accéder au classeur :



3. Générer des clés d'activation

Pour créer des clés d'activation, ouvrez le [Key Generator](#) dans XLS Padlock.

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', '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.

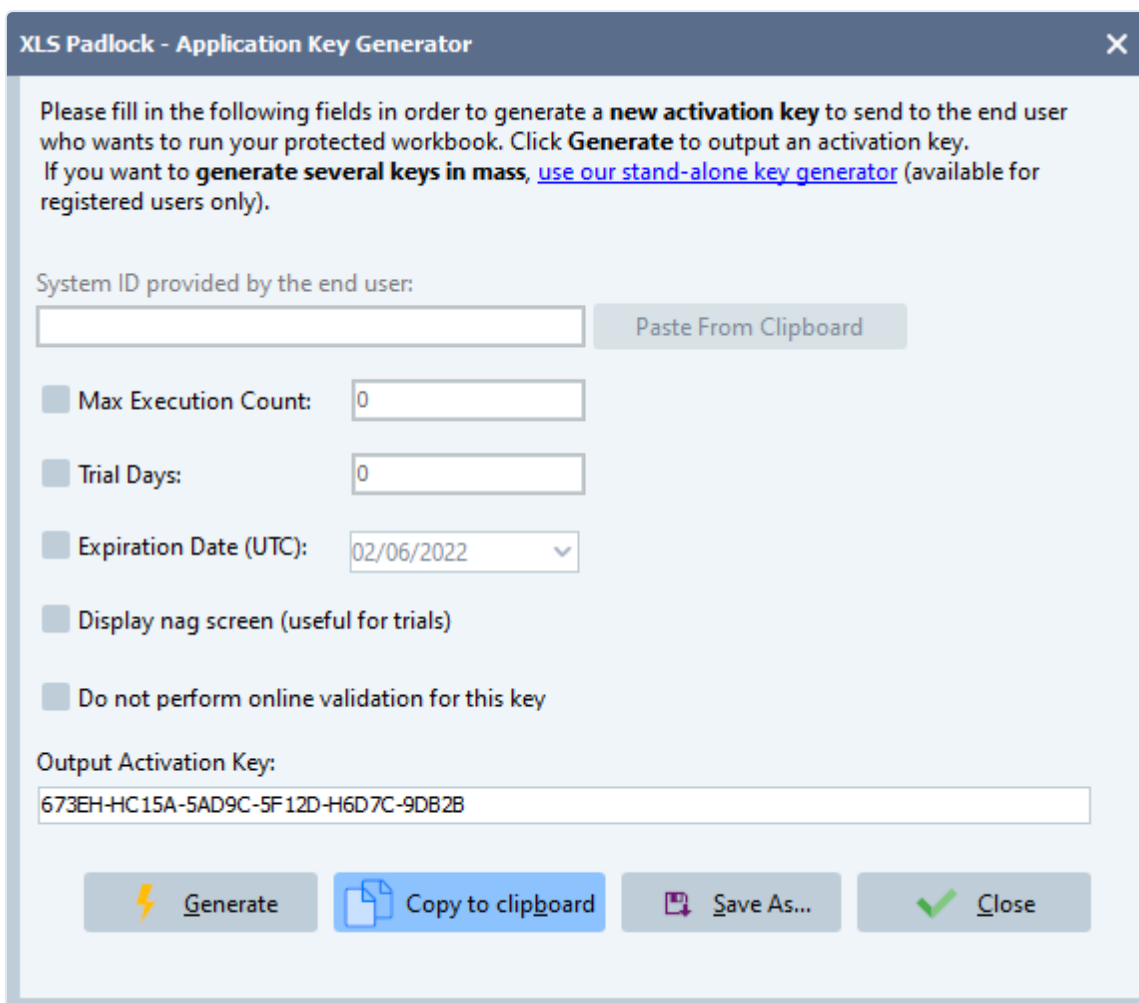
Application Master Key:

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' (highlighted with a red arrow), 'System ID Options...', and 'Clear Activation Data'.

Si vous le souhaitez, vous pouvez [définir des restrictions sur la clé](#), comme limiter le nombre d'exécutions ou définir une date d'expiration.

Cliquez sur le bouton **Generate** pour créer instantanément une clé. Vous pouvez ensuite copier cette clé dans votre presse-papiers pour l'envoyer à votre client.



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:

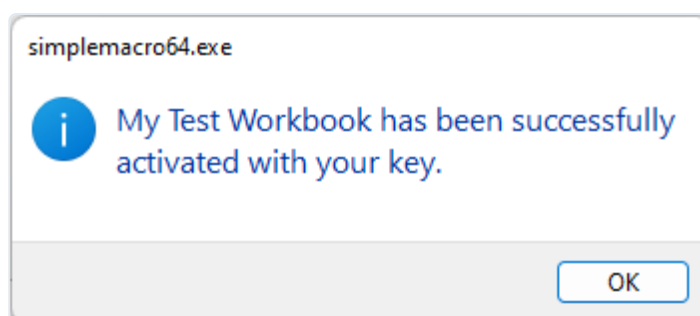
Expiration Date (UTC):

Display nag screen (useful for trials)

Do not perform online validation for this key

Output Activation Key:

Lorsque votre client saisit la clé et clique sur **Activer**, il obtient l'accès au classeur. L'application ne redemandera pas la clé, sauf si celle-ci expire.



Automatiser la génération de clés

Pour faciliter la distribution des clés, une application de [générateur de clés autonome](#) est disponible pour les clients enregistrés. Nous fournissons également des applications web gratuites permettant de générer des clés automatiquement via votre site web.

Clés d'activation verrouillées au matériel

Les **clés d'activation verrouillées au matériel** constituent un moyen puissant d'empêcher le partage occasionnel de votre classeur protégé en liant chaque licence à un ordinateur spécifique. Ce guide vous montre comment les configurer.

Tutoriel vidéo disponible

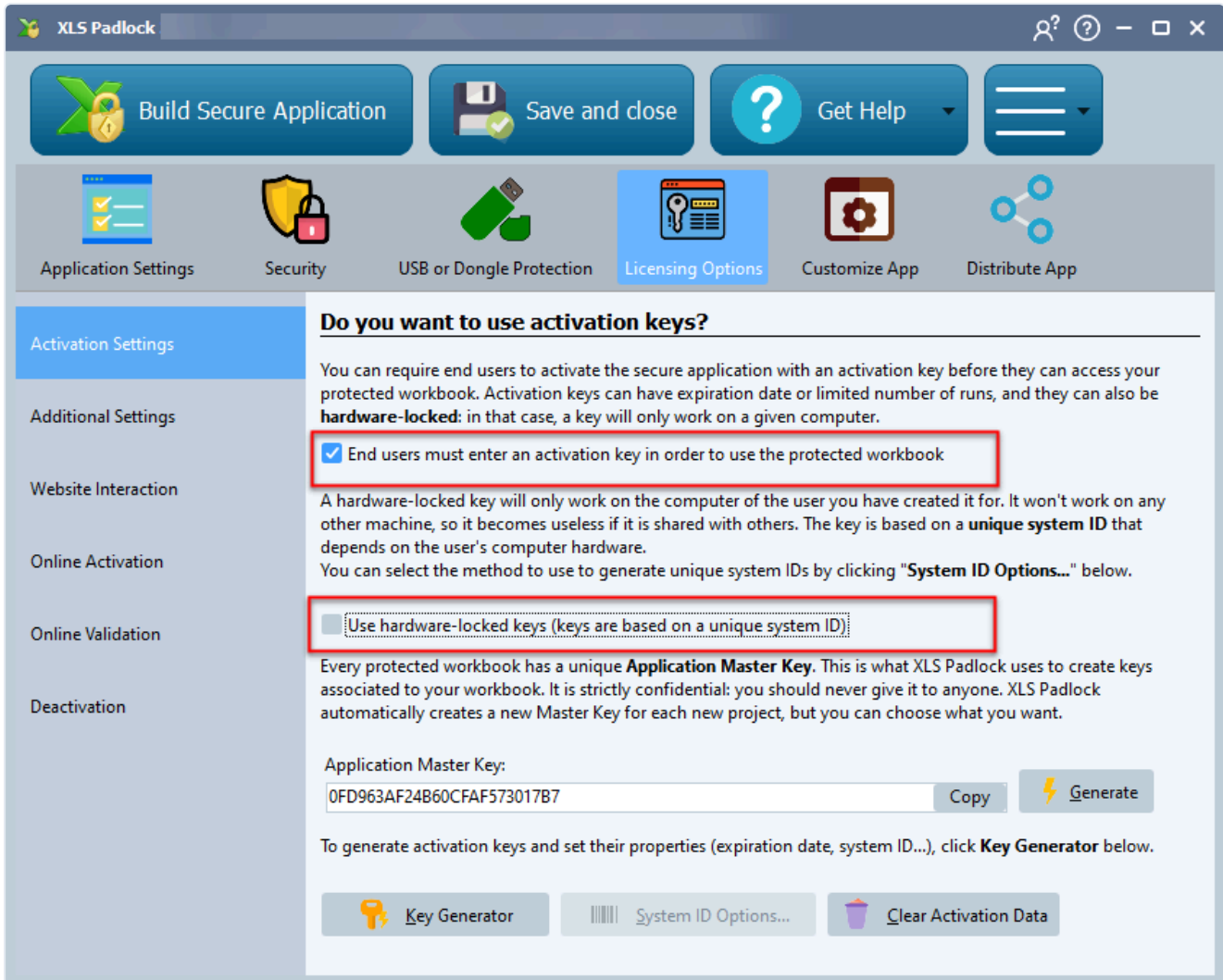
- [Regardez notre tutoriel vidéo sur le verrouillage au matériel de vos feuilles de calcul Excel](#)
- [Voir une démonstration en direct de cette fonctionnalité](#)

À propos des clés verrouillées au matériel

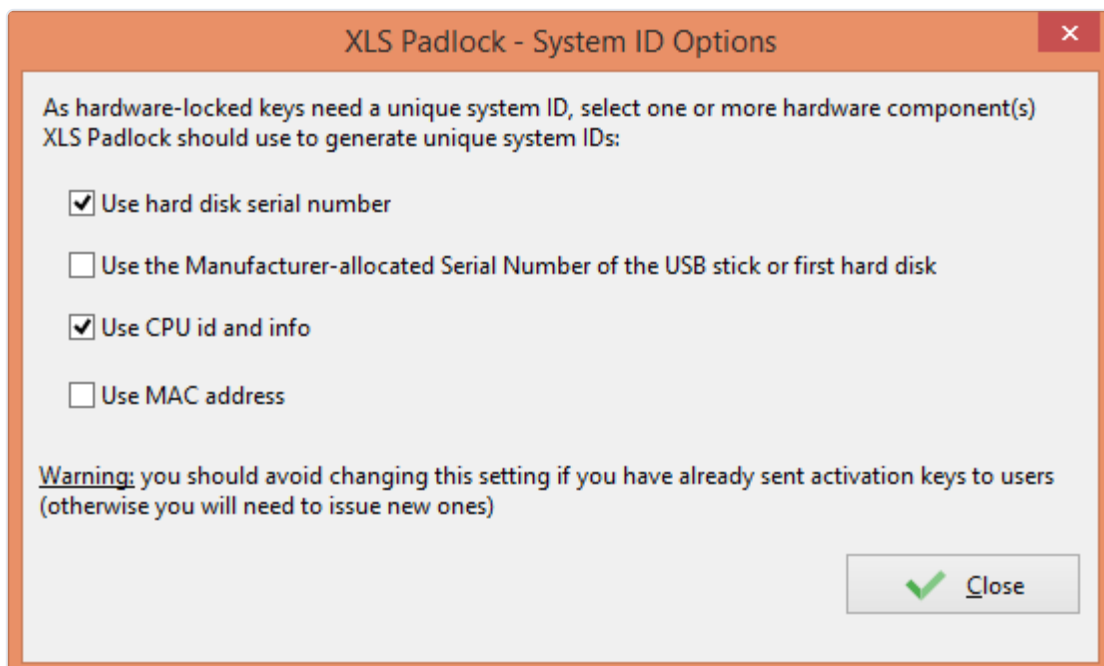
Une clé verrouillée au matériel **ne fonctionnera que sur l'ordinateur pour lequel elle a été générée**. Elle repose sur un **System ID** unique dérivé du matériel de l'ordinateur de l'utilisateur, rendant la clé inutilisable si elle est partagée avec d'autres personnes.

1. Activer les clés verrouillées au matériel

Dans les paramètres de XLS Padlock, cochez à la fois "End users must enter an activation key..." et "**Use hardware-locked keys**" (utiliser des clés verrouillées au matériel) :



Vous pouvez éventuellement configurer quels composants matériels sont utilisés pour générer le System ID en cliquant sur "**System ID Options...**". Il est important de ne pas modifier ces options après avoir commencé à déployer votre application.



Remarques sur les composants du System ID

- **CPU** : Si le CPU n'a pas d'identifiant de série intégré, ses informations générales sont utilisées. Pour du matériel identique, cela pourrait aboutir au même identifiant. Il est préférable de combiner plusieurs options matérielles.
- **MAC Address** : Si les utilisateurs disposent de plusieurs moyens de se connecter à Internet (Wi-Fi, 4G, etc.), leur adresse MAC peut changer, ce qui invaliderait leur clé.

Utiliser l'empreinte matérielle renforcée

À partir de XLS Padlock 2026.0, la boîte de dialogue System ID Options expose également une case à cocher supplémentaire : **"Use enhanced hardware fingerprint"** (utiliser l'empreinte matérielle renforcée).

Lorsque cette option est activée, l'EXE protégé utilise un calcul renforcé du System ID :

- La source de verrouillage Windows lit l'identifiant unique d'installation de Windows directement depuis le registre système au lieu du scan du volume de disque utilisé par l'algorithme historique. Cet identifiant reste stable lors des reformatages de volume et des remplacements de disque physique : seule une réinstallation complète de Windows le régénère.
- Toutes les sources matérielles sont normalisées (espaces supprimés, casse canonisée) et clairement délimitées dans le calcul de l'empreinte, éliminant une catégorie de collisions structurelles où deux machines distinctes pouvaient occasionnellement aboutir au même System ID. La nouvelle logique protège également contre les faux négatifs lorsqu'une source devient temporairement indisponible, par exemple lorsqu'un antivirus bloque les appels WMI, en gardant le System ID stable au lieu de le faire muter.
- Le hachage sous-jacent est mis à niveau vers SHA-256 à la place de l'ancien MD5, supprimant un algorithme obsolète de l'empreinte cryptographique de l'EXE protégé.

Le format affiché du System ID est inchangé : toujours 14 caractères hexadécimaux dans la forme familière `XXXX-XXXX-XXXX`, de sorte que vos utilisateurs finaux et votre flux de génération de clés ne voient aucune différence.

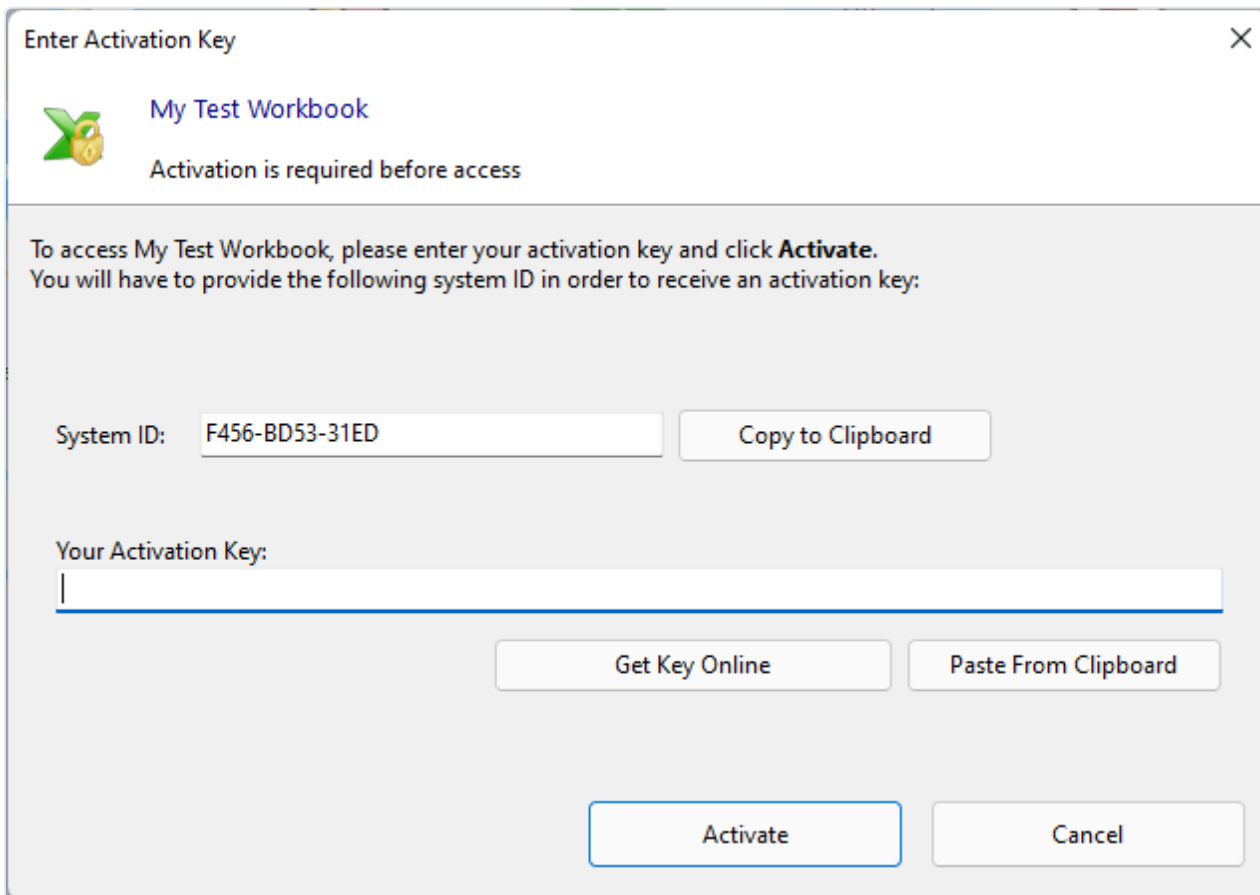
Comportement par défaut :

- **Projets existants** (enregistrés avec XLS Padlock 2025.3 ou antérieur) : l'option est **OFF** par défaut. Cela préserve toutes les clés d'activation déjà en circulation : les System IDs que vos clients ont obtenus à partir d'anciennes versions continuent de correspondre aux clés que vous avez émises.
- **Nouveaux projets** (créés avec XLS Padlock 2026.0 et ultérieur) : l'option est **ON** par défaut.

⚠ Important : modifier cette option sur un projet déjà déployé fera tourner le System ID pour chaque utilisateur final, invalidant toutes les clés d'activation que vous avez émises jusqu'à présent. Activez-la uniquement sur un nouveau projet, ou coordonnez le changement avec une réémission de la clé de chaque client.

2. Comment les utilisateurs obtiennent leur System ID

Lorsqu'un utilisateur exécute votre application pour la première fois, il sera invité à saisir une clé d'activation. Son System ID unique sera affiché dans cette fenêtre. L'utilisateur doit copier cet identifiant et vous l'envoyer.



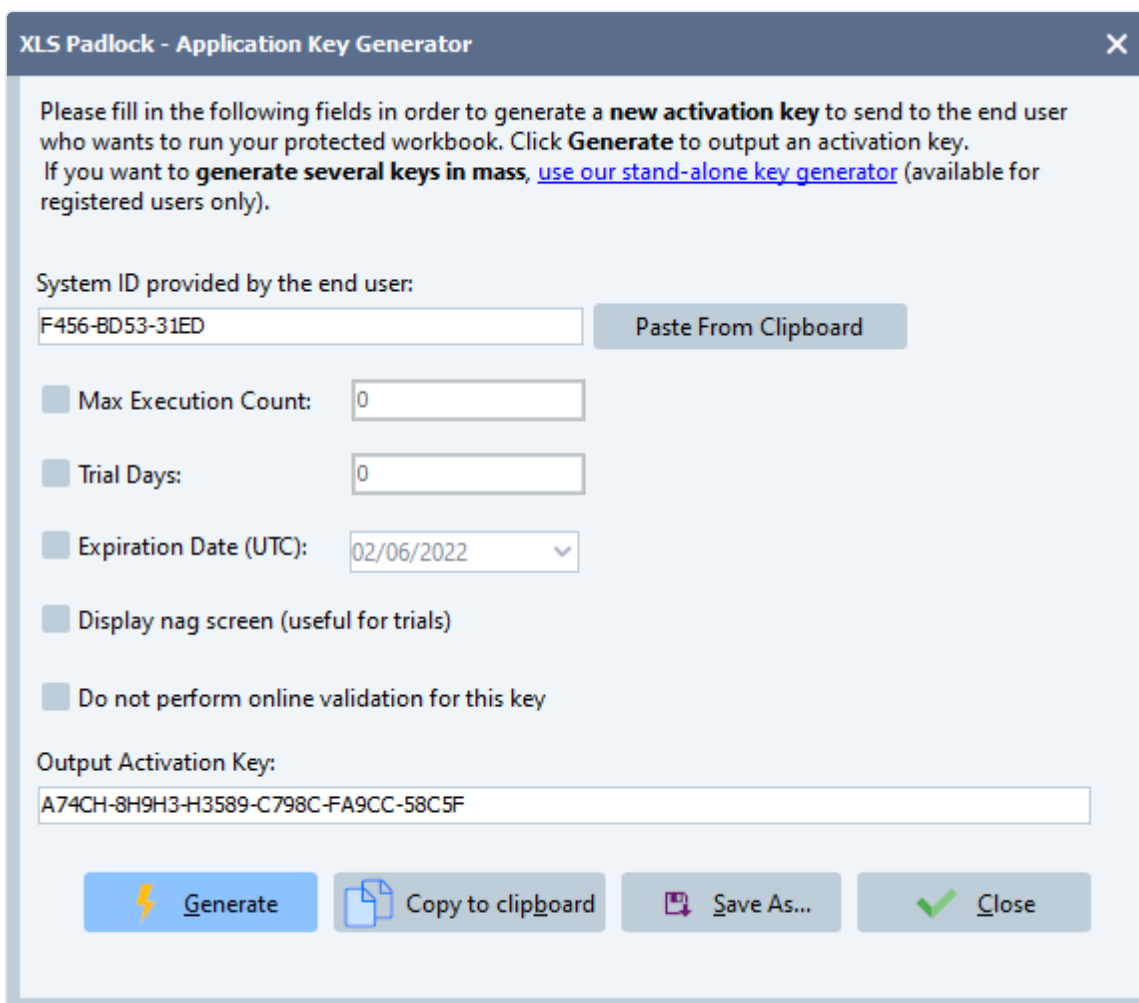
The screenshot shows a dialog box titled "Enter Activation Key" with a close button (X) in the top right corner. The dialog is for "My Test Workbook" and displays a message: "Activation is required before access". Below this, it instructs the user: "To access My Test Workbook, please enter your activation key and click **Activate**. You will have to provide the following system ID in order to receive an activation key:". The "System ID" is shown as "F456-BD53-31ED" in a text box, with a "Copy to Clipboard" button next to it. Below the System ID, there is a field labeled "Your Activation Key:" with an empty text input box. At the bottom of the dialog, there are four buttons: "Get Key Online", "Paste From Clipboard", "Activate", and "Cancel".

Automatiser la diffusion des clés

Pour éviter d'échanger manuellement les System IDs, vous pouvez configurer un bouton "[Get Key Online](#)" qui envoie l'utilisateur vers votre site web avec son System ID, ou utiliser la fonctionnalité [Activation en ligne](#) entièrement automatisée.

3\ Générer la clé d'activation

Ouvrez le [Key Generator](#) dans XLS Padlock. Collez le System ID du client dans le champ prévu à cet effet et cliquez sur **Generate**. Vous pouvez ensuite copier la clé et l'envoyer à votre client.



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:
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

Lorsque le client saisit la clé, l'application sera activée et ne redemandera plus la clé (sauf si elle a une date d'expiration). Comme la clé est liée à son System ID, elle ne fonctionnera sur aucun autre ordinateur.

👉 Vous souhaitez automatiser la création de clés d'activation verrouillées au matériel ? [En savoir plus sur l'activation en ligne.](#)

Activation en ligne

Fonctionnement

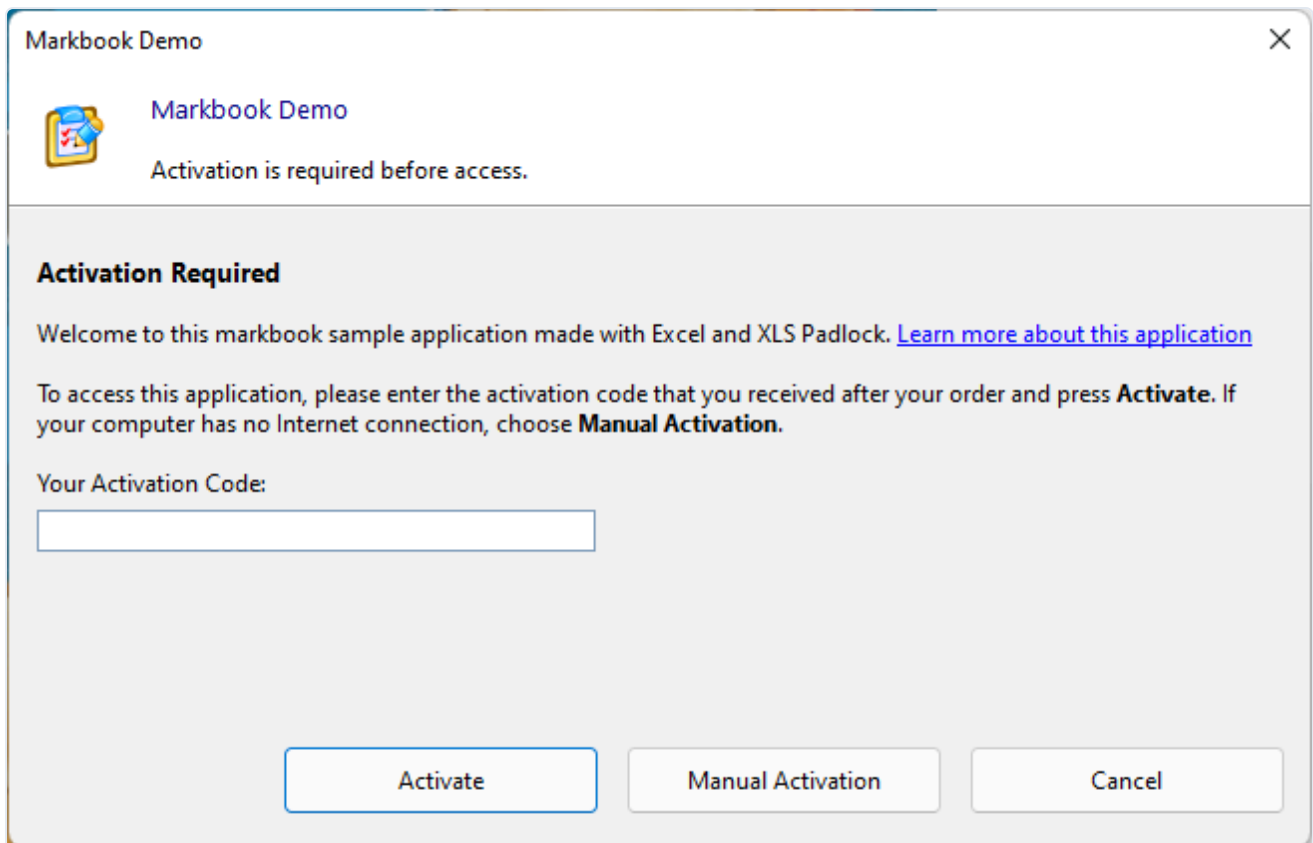
L'activation en ligne automatise la récupération des [clés d'activation](#) via Internet. L'application protégée communique avec votre serveur web pour télécharger directement une clé d'activation, ce qui évite aux utilisateurs de devoir la saisir manuellement.

Kit côté serveur requis

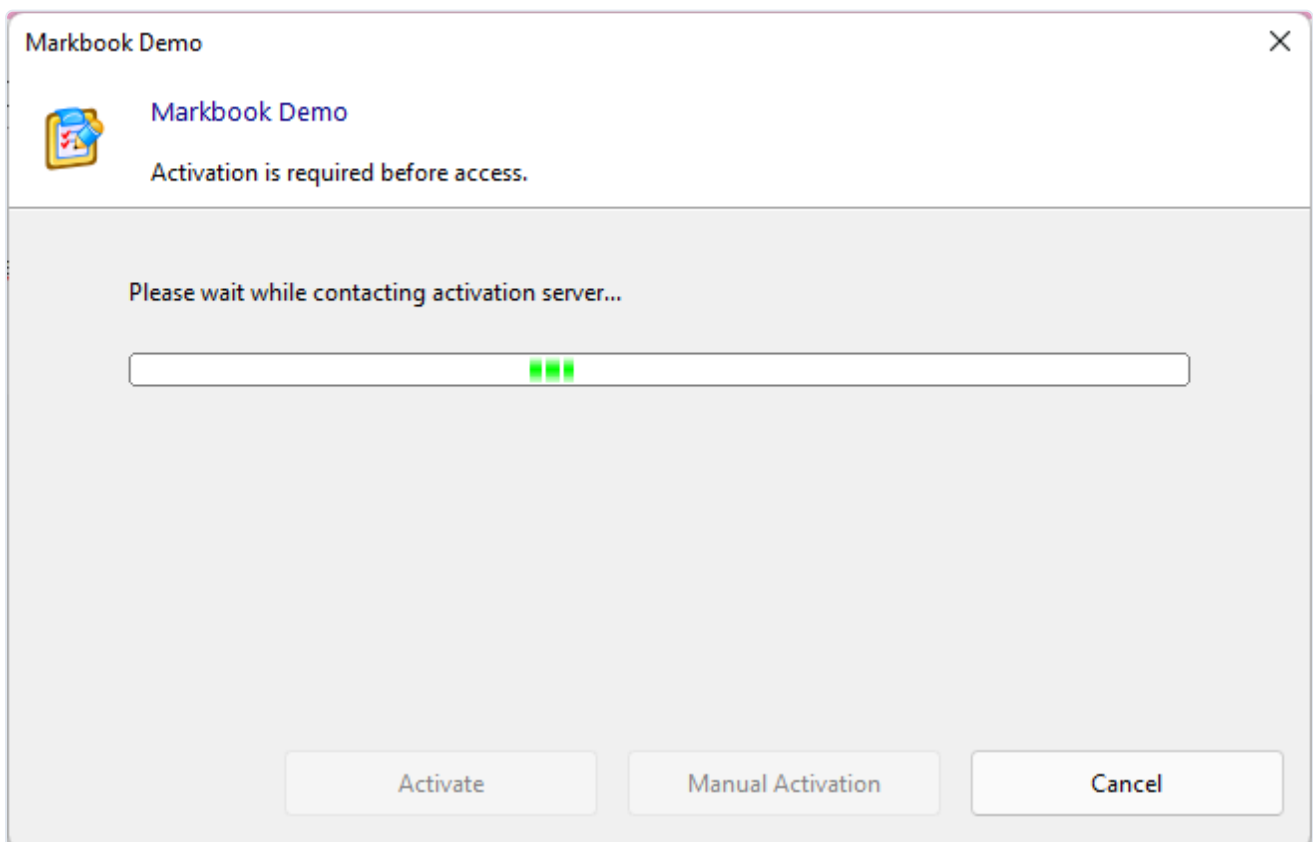
Pour utiliser l'activation en ligne, vous devez installer l'un de nos kits côté serveur sur votre serveur web, par exemple le XLS Padlock Activation Kit, le WooCommerce Integration Kit ou le FastSpring Subscription kit. Vous pouvez télécharger ces kits depuis votre [page de compte](#).

Si vous optez pour le protocole d'activation 2026 (signature des réponses Ed25519, voir ci-dessous), assurez-vous que votre kit est en version 2026.0 ou ultérieure. Le kit détecte automatiquement le protocole à chaque requête, de sorte que les classeurs hérités continuent de fonctionner avec la même installation du kit.

👉 Lorsqu'un utilisateur final démarre l'application, une boîte de dialogue apparaît pour lui demander de procéder à l'activation. Cette boîte de dialogue remplace la boîte de dialogue standard « Enter Activation Key » et est entièrement personnalisable. Vous pouvez ajouter vos propres champs pour collecter des données auprès de l'utilisateur (comme un numéro de commande ou une adresse e-mail), qui seront envoyées à votre serveur web. Votre serveur valide ensuite ces informations et, en cas de succès, renvoie une clé d'activation à l'application.



Lorsque l'utilisateur clique sur Activer, les données sont envoyées au serveur d'activation :



Après une activation réussie, un message de confirmation s'affiche et l'application redémarre. En cas d'erreur, une boîte de message apparaît, permettant à l'utilisateur de réessayer.

Configuration

👉 Pour activer l'activation en ligne, vous devez configurer les options suivantes :

Base Activation URL

Indiquez l'URL complète du kit d'activation installé sur votre serveur. Par exemple, si vous avez installé le kit dans un sous-dossier nommé « activation », l'URL serait

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

Utilisez HTTPS

Les connexions sécurisées via TLS/SSL sont prises en charge. Vous devez toujours utiliser des URL commençant par `https://`.

⚠️ Laissez le champ vide si vous ne souhaitez **pas** utiliser l'activation en ligne.

Identification du client : clé de sécurité ou paire de clés Ed25519

À partir de XLS Padlock 2026.0, vous pouvez choisir entre deux régimes de protocole d'activation via l'option **Compatibility mode for pre-2026 activation kits** (mode de compatibilité pour les kits d'activation antérieurs à 2026) :

- **Compatibility mode coché** (par défaut, recommandé si votre serveur d'activation est antérieur à la version 2026) : l'ancien champ GUID **Security Private Key** identifie votre application auprès du kit d'activation. Le protocole est basé sur des lignes avec des réponses non signées. C'est le comportement de toutes les versions de XLS Padlock antérieures à 2026.0.
- **Compatibility mode décoché** (nécessite la version 2026 du XLS Padlock Activation Kit, du WooCommerce Integration Kit ou du FastSpring Subscription Kit) : l'ancien champ est masqué et une **Ed25519 keypair** (paire de clés Ed25519) propre au projet prend le relais. Cliquez sur le bouton **Generate keypair** (générer la paire de clés) sur la page Online Activation pour produire une nouvelle paire de clés. La clé publique est intégrée dans le classeur protégé ; la clé privée vous est présentée dans une boîte de dialogue à affichage unique afin que vous puissiez la coller dans la configuration de votre kit d'activation (`xlspadlocksignkey` dans `config.ini`).

Si vous perdez la clé privée, cliquez sur **Show config.ini snippet** sur la même page pour la réafficher. La clé est conservée dans votre fichier de projet `.xplp`. Traitez le fichier `.xplp` comme sensible : ne le placez pas sous contrôle de version, ne l'envoyez pas par e-mail pour obtenir de l'assistance.

Le protocole 2026 utilise une enveloppe de requête JSON et vérifie une signature Ed25519 détachée sur chaque réponse du serveur d'activation. Cela protège vos clients contre les fausses réponses « activées » ou les messages d'erreur de phishing qu'un attaquant au niveau du réseau pourrait autrement injecter en contournant TLS (proxy d'entreprise avec injection de CA, antivirus malveillant, certificat racine compromis).

Allow Manual Activation if No Internet Connection

Certains utilisateurs peuvent ne pas disposer d'une connexion Internet active. Pour leur permettre d'activer manuellement, activez l'option « **Allow Manual Activation if No Internet Connection** ».

Cette méthode manuelle fonctionne exactement comme les clés d'enregistrement standard. Dans ce cas, vous devez être prêt à traiter les demandes d'activation des utilisateurs hors ligne.

Avertissement

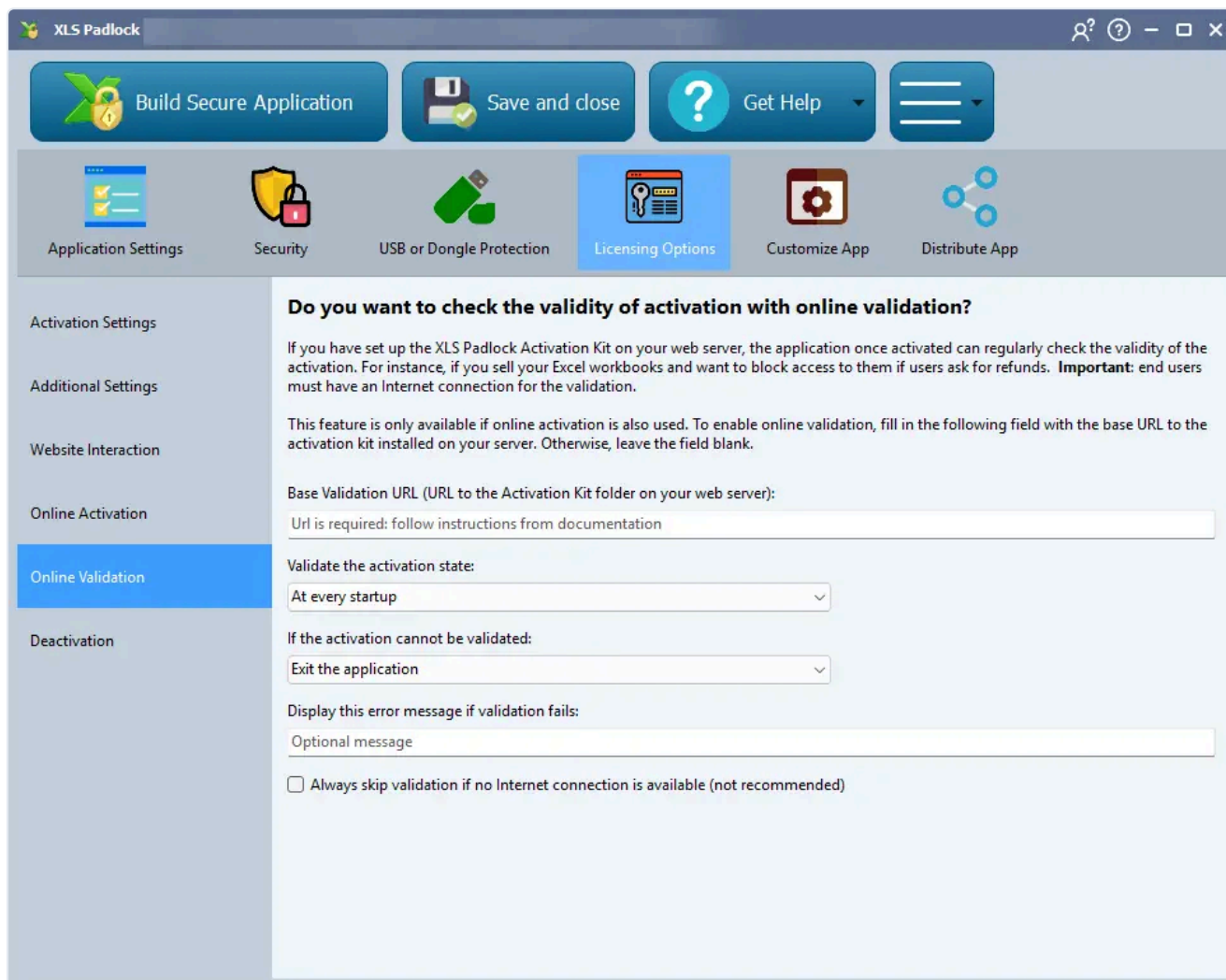
Si vous acceptez les activations manuelles, veillez à générer des clés d'activation qui **n'effectuent pas de validation en ligne** (option dans le [générateur de clés](#)). Sinon, la validation échouera !

Voir aussi

- [Comment vendre des classeurs Excel avec des abonnements](#)
- [Afficher un bouton « Purchase Online » sur l'écran de rappel](#)

Validation en ligne

La validation en ligne offre un **contrôle à distance** sur vos applications activées. Lorsqu'elle est utilisée avec l'un de nos kits d'activation côté serveur (tels que le XLS Padlock Activation Kit, le WooCommerce Kit ou le [kit d'abonnement FastSpring](#)), votre application peut vérifier périodiquement auprès de votre serveur que son activation reste valide.



Avertissement

Cette fonctionnalité nécessite que l'[activation en ligne](#) soit d'abord activée et configurée.

Cela est utile dans plusieurs cas de figure :

- Vous pouvez désactiver à distance une application si un client demande un remboursement.
- Vous pouvez imposer un accès basé sur un abonnement, où l'application cesse de fonctionner si l'abonnement expire.

Comment cela fonctionne

- La validation en ligne utilise le même jeton unique qui a été généré lors de l'activation en ligne initiale pour identifier l'appareil du client.
- La clé d'activation elle-même n'est jamais transmise sur internet pendant la validation.
- Une connexion internet active est requise pour que la vérification de validation puisse avoir lieu.

Configuration

👉 Vous devez configurer la **Base Validation URL** pour qu'elle pointe vers votre script de validation côté serveur. Il s'agit généralement de l'URL du XLS Padlock Activation Kit, du XLS Padlock WooCommerce Integration Kit ou du kit d'abonnement FastSpring sur votre serveur web. Par exemple, si vous avez installé le kit d'activation dans un sous-dossier nommé « activation », l'URL serait

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

Utilisez HTTPS

Les connexions sécurisées utilisant TLS/SSL sont prises en charge. Vous devez toujours utiliser des URL commençant par `https://` .

⚠ Laissez le champ vide si vous **ne** souhaitez **pas** utiliser la validation en ligne.

Fréquence de validation

Choisissez le moment où l'application doit effectuer une validation : à chaque démarrage, de façon aléatoire, tous les X jours ou toutes les X exécutions. Vous devez préciser la valeur de X lorsque cela est requis.

Avertissement

Après l'activation, une validation initiale est requise au prochain lancement. Après cette première vérification, l'application respectera la fréquence de validation sélectionnée.

Action en cas d'échec de la validation

Définissez ce que l'application doit faire si la validation en ligne échoue : * **Exit the application** : l'application se fermera immédiatement. * **Blacklist activation key** : la clé actuelle sera invalidée, et l'utilisateur sera invité à en saisir une nouvelle. Si l'utilisateur saisit à nouveau la même clé, la validation sera de nouveau tentée. * **Do nothing** : l'application continuera de fonctionner. Vous pouvez utiliser les extensions VBA de XLS Padlock pour vérifier l'état de validation et mettre en oeuvre une logique personnalisée.

Si l'activation ne peut pas être validée

Vous pouvez afficher un message d'erreur personnalisé à l'utilisateur si la validation échoue, lui indiquant la marche à suivre.

Ignorer la validation si hors ligne

Vous pouvez autoriser l'application à ignorer la vérification de validation si aucune connexion internet n'est détectée. Cela n'est généralement pas recommandé pour des raisons de sécurité.

Avertissement

Si cette option est désactivée, les utilisateurs doivent disposer d'une connexion internet active pour le processus de validation. Dans le cas contraire, la validation échouera.

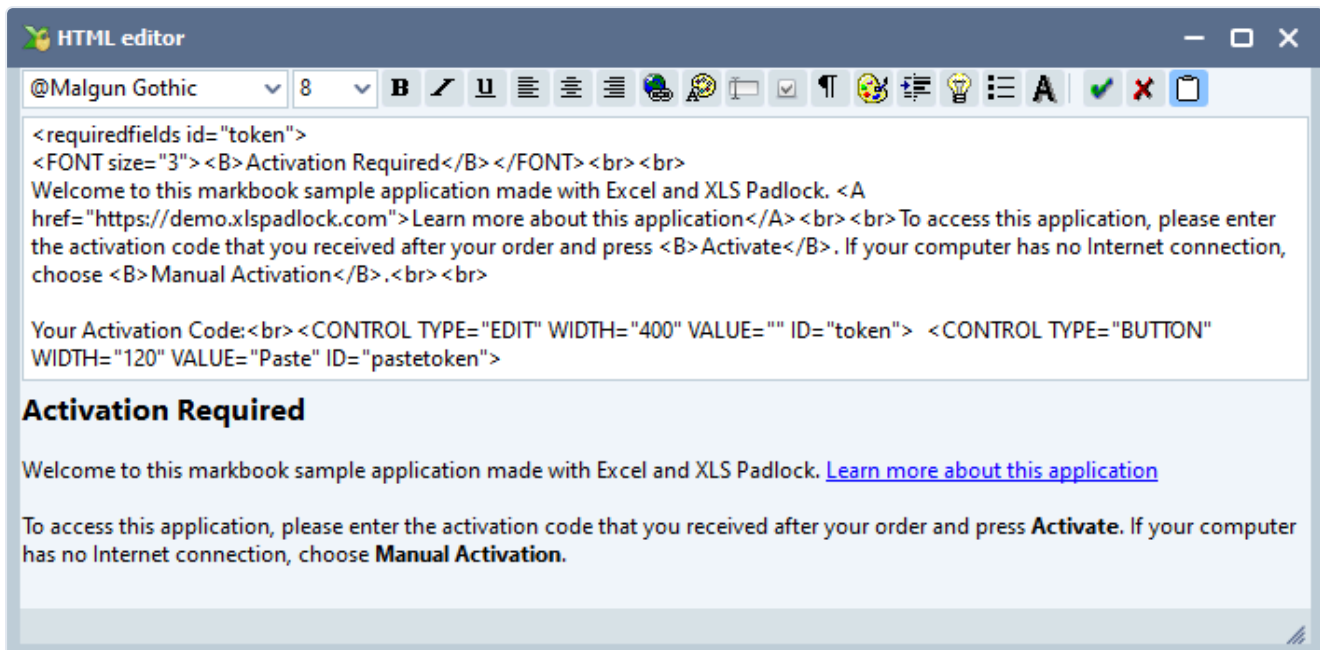
VBA : récupérer les informations d'abonnement/licence

Après une validation en ligne réussie, vous pouvez récupérer des données personnalisées envoyées par votre serveur à l'aide de l'appel VBA `XLSpadlock.PLEvalVar("ValidationAddServerData")`. Cela est utile pour transmettre des informations spécifiques à l'abonnement au classeur.

Éditeur du formulaire d'enregistrement

L'éditeur de formulaire d'enregistrement vous permet de personnaliser le texte qui apparaît dans la [boîte de dialogue d'activation en ligne](#). L'éditeur prend en charge le HTML de base, vous pouvez donc utiliser des balises comme ``, `
`, ``, etc.

XLS Padlock fournit un éditeur HTML de base avec un aperçu en direct du texte de la boîte de dialogue :



Contrôles personnalisés

Vous pouvez ajouter des champs personnalisés pour demander des informations supplémentaires aux utilisateurs, qui seront envoyées à votre serveur web. Vous pouvez également ajouter un bouton pour coller du texte dans le champ du jeton.

- **Paste Button** : `<CONTROL TYPE="BUTTON" WIDTH="120" VALUE="Paste" ID="pastetoken">`
- **Text Input** : `<CONTROL TYPE="EDIT" WIDTH="400" VALUE="" ID="token">`

Champs obligatoires

Vous pouvez rendre des champs obligatoires en listant leurs ID dans une balise `<requiredfields>`. Par exemple, pour rendre obligatoires les champs « token » et « name » :

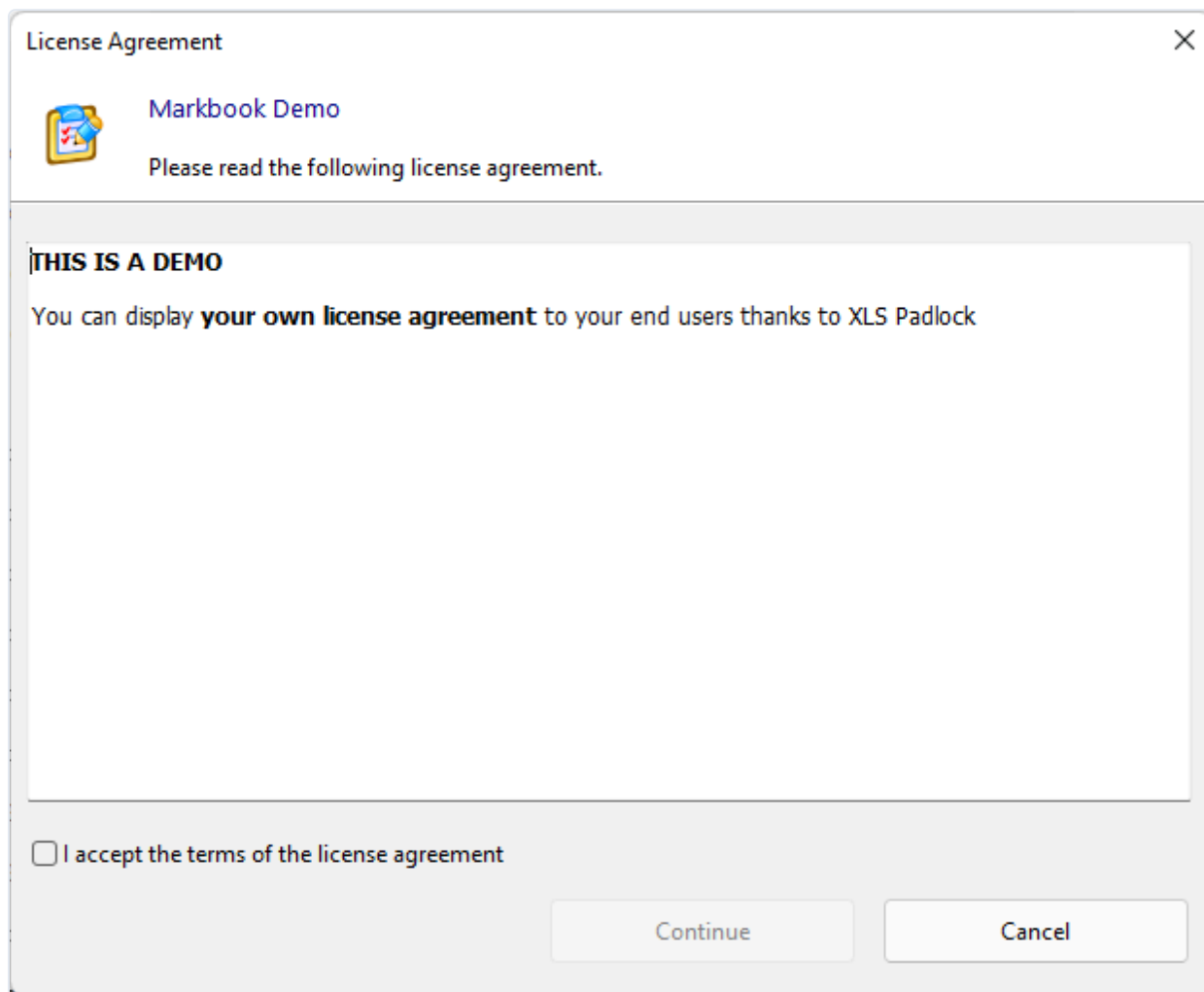
```
<requiredfields id="token,name">
```

Exemple complet

Afficher le contrat de licence

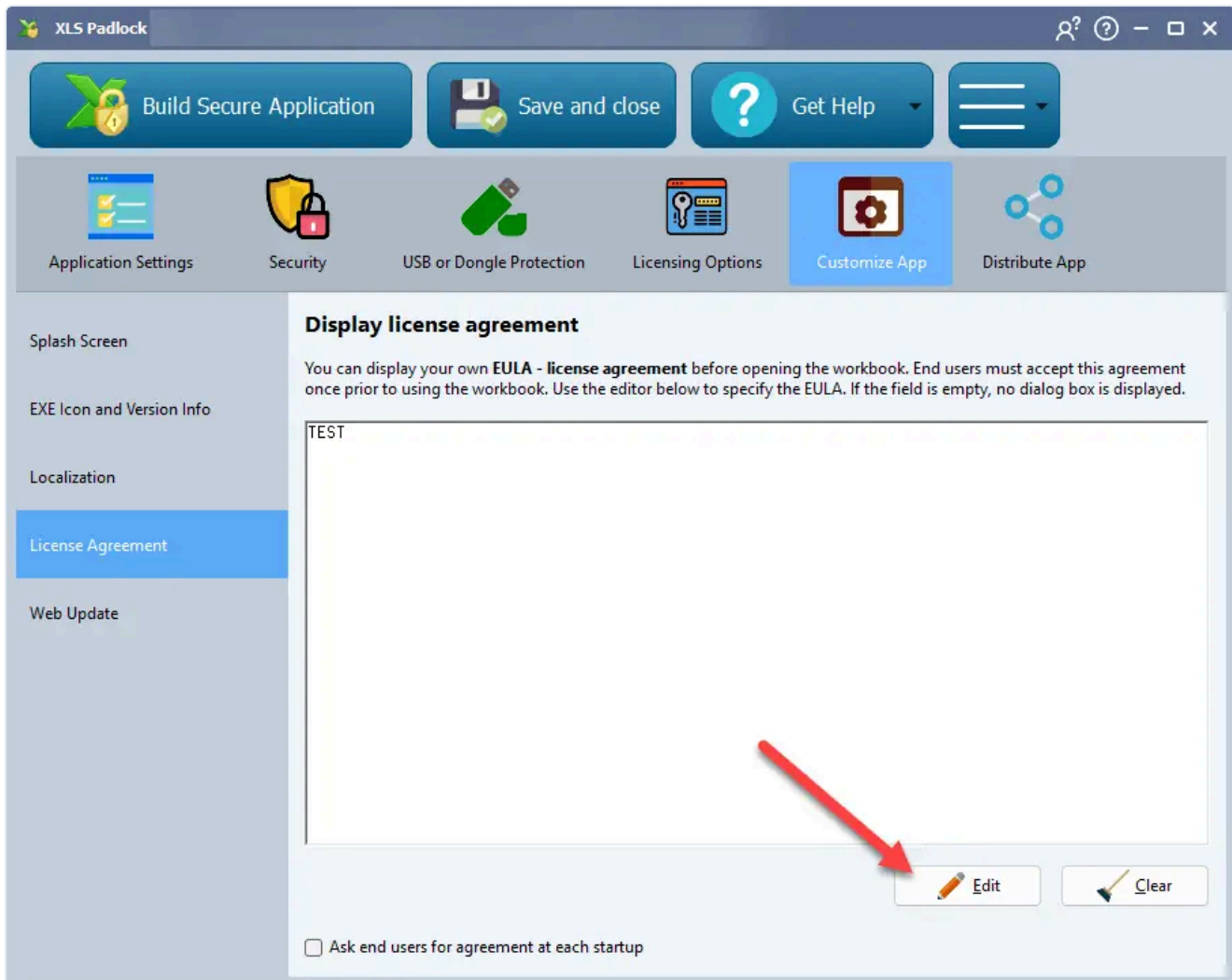
Vous pouvez exiger des utilisateurs qu'ils acceptent un **End User License Agreement (EULA)** (contrat de licence utilisateur final) avant de pouvoir ouvrir votre classeur protégé.

Le classeur compilé affichera une boîte de dialogue comme illustré ci-dessous :



Lorsque l'utilisateur coche « I accept the terms of the license agreement », le bouton « Continue » devient actif, lui permettant de poursuivre.

👉 Pour ajouter votre EULA, cliquez sur **Edit** afin d'ouvrir un éditeur de texte enrichi sur cette page :



Vous pouvez charger le contenu à partir d'un fichier RTF ou coller du texte directement depuis une application comme Microsoft Word.

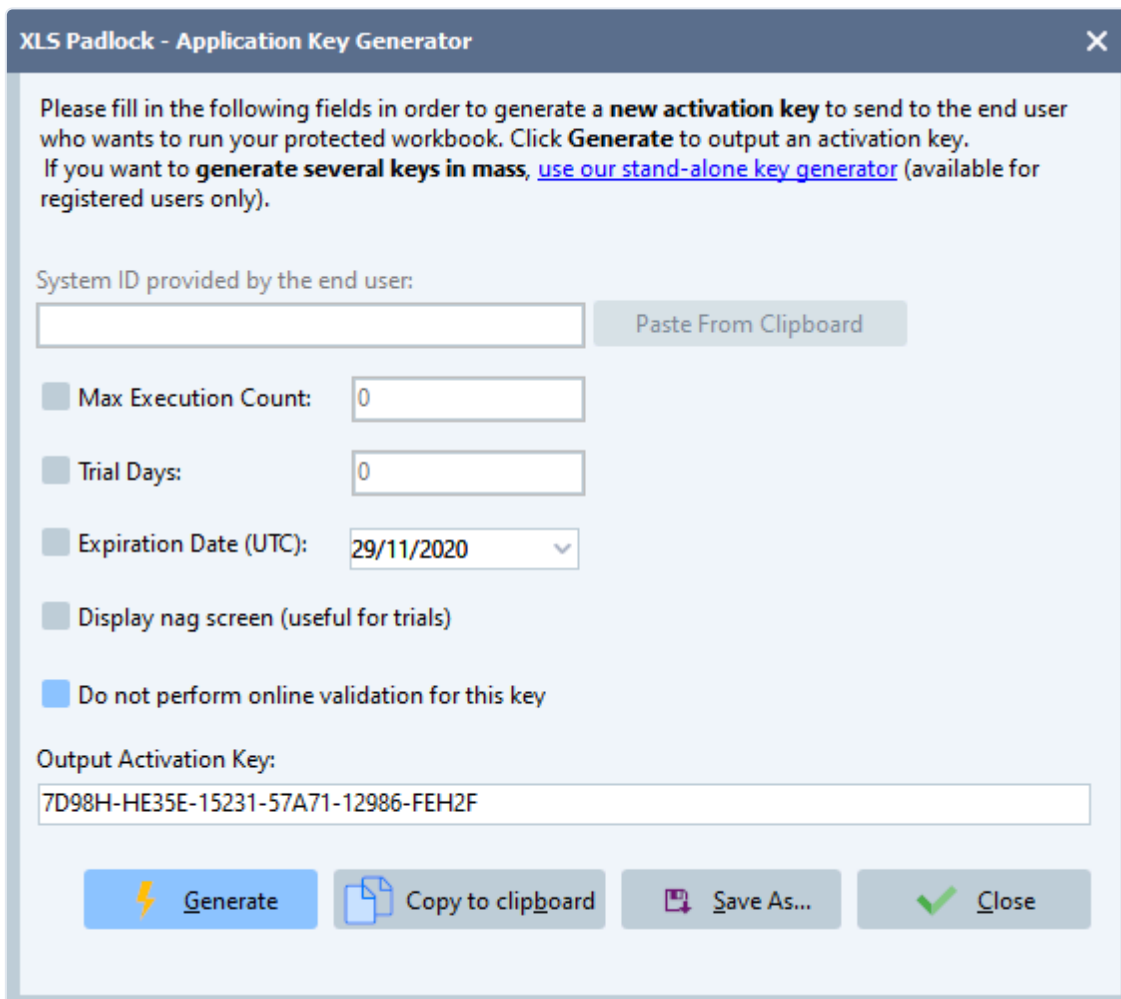
REMARQUE

Si le contenu de l'EULA est laissé vide, aucune boîte de dialogue ne sera affichée.

i Pour afficher l'EULA à chaque démarrage de l'application, activez l'option **Ask end users for agreement at each startup**. Par défaut, les utilisateurs n'ont besoin de l'accepter qu'une seule fois.

Générateur de clés (portable et serveur)

XLS Padlock inclut un **Key Generator** (générateur de clés) intégré pour créer des clés d'activation pour vos applications de classeur.



- **Pour les clés d'activation simples**, cliquez simplement sur le bouton **Generate** pour créer instantanément une clé.
- **Pour les clés verrouillées au matériel**, saisissez le System ID fourni par votre utilisateur, puis cliquez sur **Generate**. La clé obtenue sera liée à cet ordinateur spécifique.

Vous pouvez copier la clé générée dans le presse-papiers pour l'envoyer à votre client, ou l'enregistrer sous forme de fichier `.txt` pour vos archives.

Les clés sont spécifiques au classeur

Le générateur de clés crée des clés d'activation uniquement pour le projet de classeur actuellement ouvert. Les clés générées pour un classeur ne fonctionneront pas avec un autre, sauf s'ils partagent la même [Application Master Key](#).

SDK du générateur de clés

Nous proposons un Key Generator SDK gratuit (une application web basée sur PHP) que vous pouvez installer sur votre propre site web. Ce SDK vous permet d'automatiser la génération et la distribution des clés d'activation pour vos classeurs.

Contrairement au [générateur de clés intégré](#), qui crée les clés manuellement dans XLS Padlock, ce SDK s'exécute sur votre propre serveur pour émettre les clés automatiquement.

Pour une solution de licence entièrement automatisée, envisagez d'associer le SDK à la [fonctionnalité d'activation en ligne](#).

👉 Vous pouvez télécharger le Key Generator SDK depuis votre [page de compte](#).

Générateur de clés autonome

XLS Padlock propose un générateur de clés autonome pour les clients enregistrés. Cet outil vous permet de [créer des clés d'activation](#) pour vos classeurs protégés sans avoir à ouvrir Excel ni le complément 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

Pour une gestion plus simple, le générateur de clés autonome vous permet également d'enregistrer les paramètres sous forme de **profiles** (profils).

👉 Vous pouvez télécharger le générateur de clés autonome depuis votre [page de compte](#).

Restrictions sur les clés

Lorsque vous générez des [Activation Keys](#) avec XLS Padlock, vous pouvez **appliquer diverses restrictions** à l'aide des options suivantes :

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)

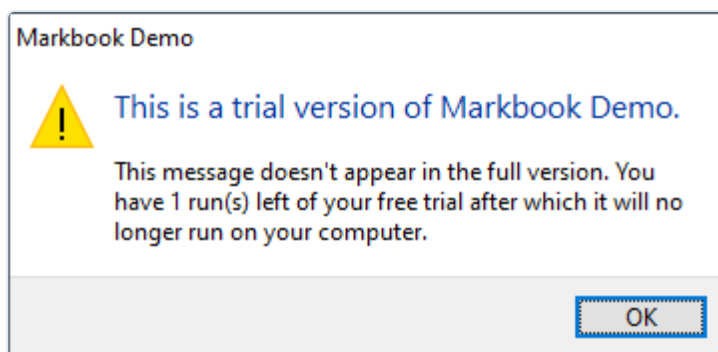
Do not perform online validation for this key

Output Activation Key:

👉 Limitez le nombre de fois où votre application peut être exécutée en activant "**Max Execution Count**" et en saisissant le nombre souhaité.

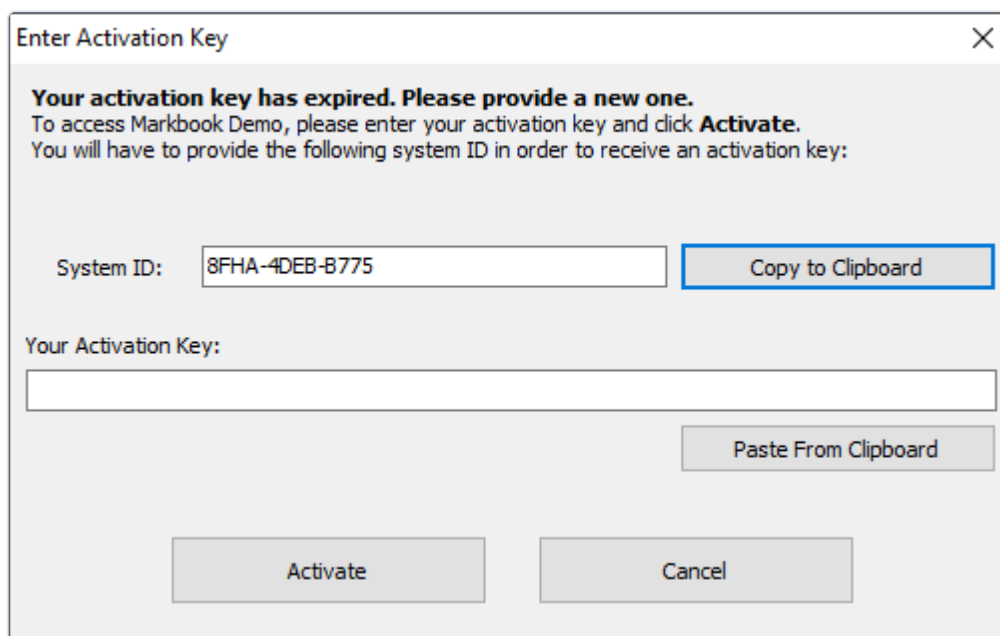
👉 Faites expirer la clé après un nombre de jours précis (activez "**Trial Days**") ou à une date précise (activez "**Expiration Date**").

👉 Affichez une boîte de dialogue de rappel (ou "nag screen") au démarrage en activant "**Display Nag Screen**". Ceci est [utile pour les versions d'essai](#), car cela informe l'utilisateur du nombre de jours ou d'exécutions restants.



Enfin, si vous utilisez la [validation en ligne](#) et que vous devez exempter une clé spécifique de cette vérification (par exemple, pour un utilisateur dans un environnement hors ligne), vous pouvez activer "**Do not perform online validation for this key**". Ceci est recommandé uniquement lorsque cette option est combinée avec des clés verrouillées sur le matériel (Hardware Lock).

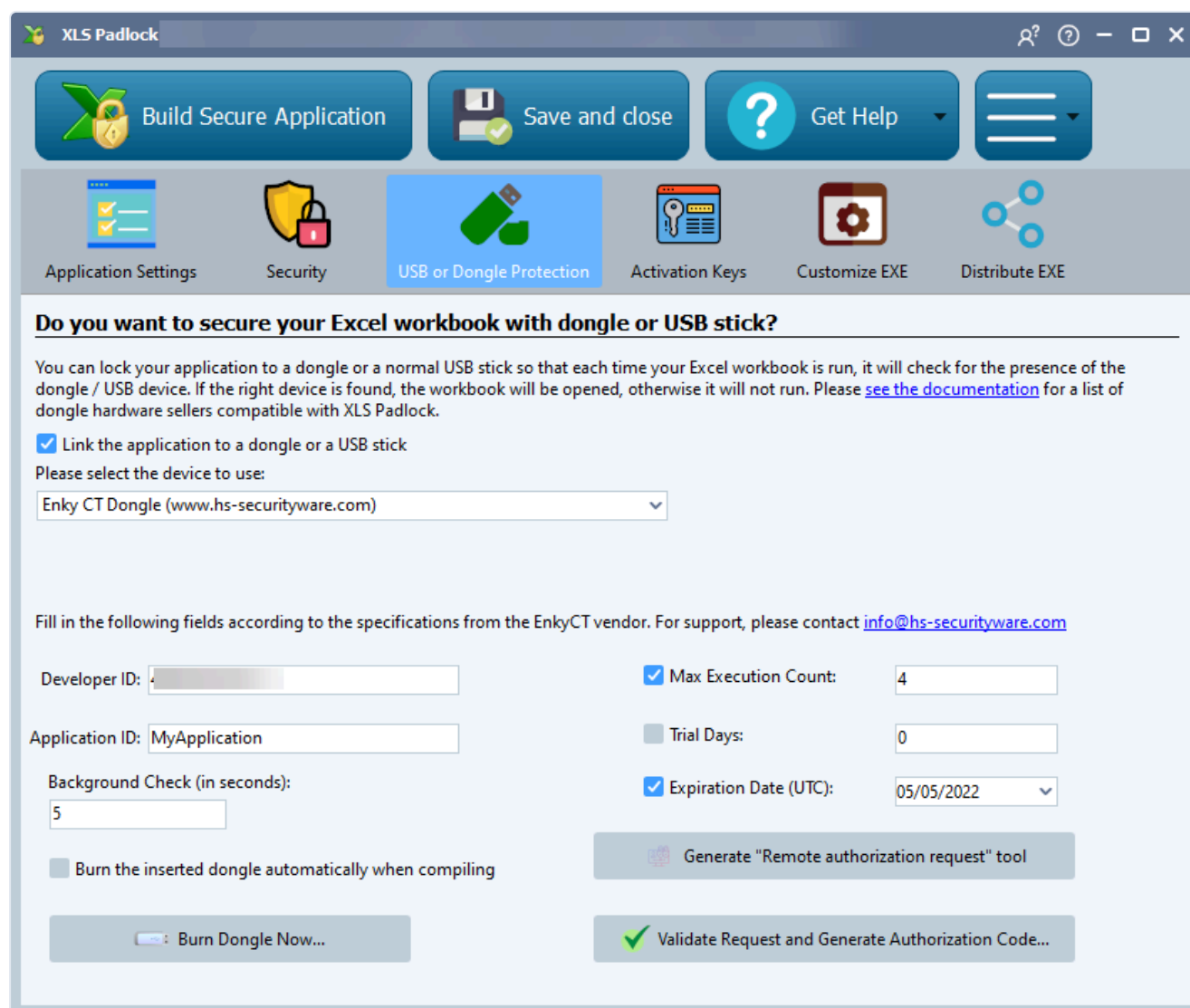
Lorsque la limite d'une clé (date, nombre d'exécutions ou jours d'essai) est atteinte, l'utilisateur sera invité à saisir une nouvelle clé pour continuer à utiliser l'application :



Protection par USB ou dongle

Avec XLS Padlock, vous pouvez verrouiller vos classeurs Excel sur une clé USB ou sur un dongle de sécurité dédié. Cela signifie que **l'application du Compiled Workbook ne fonctionnera pas si le périphérique USB correct n'est pas inséré**. L'application vérifiera la présence du périphérique au démarrage et peut aussi la vérifier régulièrement pendant l'exécution.

Pour utiliser cette protection, cochez l'option « **Link the application to a dongle or a USB stick** » sur la page "USB Protection" et sélectionnez le modèle de périphérique que vous possédez.



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 active, displaying the following content:

Do you want to secure your Excel workbook with dongle or USB stick?

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.

Link the application to a dongle or a USB stick

Please select the device to use:

Enky CT Dongle (www.hs-securityware.com) ▼

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

Developer ID: <input type="text"/>	<input checked="" type="checkbox"/> Max Execution Count: <input type="text" value="4"/>
Application ID: <input type="text" value="MyApplication"/>	<input type="checkbox"/> Trial Days: <input type="text" value="0"/>
Background Check (in seconds): <input type="text" value="5"/>	<input checked="" type="checkbox"/> Expiration Date (UTC): <input type="text" value="05/05/2022"/>

Burn the inserted dongle automatically when compiling

XLS Padlock prend en charge plusieurs types de protection matérielle pour verrouiller votre application Excel sur un périphérique physique.

Dongle Enky CT

Selon le fournisseur, Enky CT est un dongle d'horloge temporelle à base de carte à puce 32 bits, simple d'utilisation et flexible, principalement utilisé pour la protection logicielle et la limitation dans le temps.

Pour obtenir de l'aide, veuillez contacter [info@hs-securityware.com] ou visiter [<https://hs-securityware.com>].

Comment utiliser le dongle Enky CT

Saisissez le **Developer ID** fourni par HS Securityware (il est sensible à la casse).

L'**Application ID** peut être n'importe quelle valeur permettant d'identifier le dongle, et il doit être lié à l'application que vous créez.

Background Check

L'application vérifie la présence du dongle au démarrage et peut aussi la vérifier régulièrement pendant l'exécution. Vous pouvez définir la fréquence de ces vérifications ; par exemple, une valeur de 20 signifie que l'application vérifiera la présence du dongle toutes les 20 secondes. Si le dongle n'est pas trouvé, un nag screen apparaît, obligeant l'utilisateur à insérer le dongle dans les 15 secondes, faute de quoi l'application se fermera.

Set restrictions on the application

Vous pouvez limiter le nombre de fois où votre application peut être exécutée en définissant un **Max Execution Count**. Vous pouvez aussi faire expirer votre application après un nombre de jours donné (**Trial Days**) ou à une date précise (**Expiration Date**).

Fonctions de mise à jour à distance du dongle

Ce modèle de dongle vous permet de mettre à niveau à distance des dongles déjà en possession de vos clients. Par exemple, vous pouvez prolonger une date d'expiration ou ajouter des exécutions supplémentaires. Cette fonction de mise à jour à distance repose sur un système de code de requête et d'autorisation.

Comment mettre à jour à distance le dongle d'un client :

1. Tout d'abord, générez l'outil "remote authorization request tool" depuis XLS Padlock et envoyez à votre client les fichiers EXE et DAT obtenus.

Avertissement

Ces deux fichiers (`.EXE` et `.DAT`) doivent rester dans le même dossier pour que l'outil fonctionne.

2. Le client exécute l'outil pour générer un **request code**, qu'il vous envoie.
3. Dans XLS Padlock, choisissez « Validate Request and Generate Authorization Code », collez le request code, puis cliquez sur "Validate Request Code". Vous pouvez alors voir l'état actuel du dongle et définir de nouvelles limites (nombre d'utilisations ou nouvelle date d'expiration).
4. Générez l'**authorization code** et renvoyez-le au client. Il utilisera le même outil pour saisir le code et appliquer la mise à jour à son dongle.

Dongle Enky LC

Pour obtenir de l'aide, veuillez contacter [info@hs-securityware.com] ou visiter [<https://hs-securityware.com>].

L'Enky LC est un dongle HID économique et sans pilote pour la protection logicielle.

Comment utiliser le dongle Enky LC2

1. Saisissez le **Developer ID** que vous avez reçu de HS-Security Ware.
2. Saisissez un **Product ID** unique pour votre classeur. Cela garantit que seuls les dongles ayant le bon Product ID seront acceptés.

Étapes de configuration d'un dongle Enky LC2

Pour lier un dongle à votre application, vous devez d'abord le "graver" à l'aide du bouton « **Burn Dongle Now** » dans XLS Padlock. Cela configure le dongle avec vos identifiants. Ce processus peut aussi se produire automatiquement lorsque vous compilez votre application si un dongle compatible est branché.

Background Check

Vous pouvez activer une vérification en arrière-plan (background check) pour garantir que le dongle reste branché pendant l'exécution de l'application. Vous pouvez définir l'intervalle (en secondes) selon lequel la vérification est effectuée.

Protection par clé USB générique

Vous pouvez verrouiller votre application sur une ou plusieurs clés USB génériques en fonction de leur ID de fabricant unique.

- Tout d'abord, vous devez saisir un **Application Secret ID**. Celui-ci est combiné à l'ID de la clé USB pour générer un hachage unique.
- Lorsque l'application se lance, elle analyse tous les lecteurs USB. Si un lecteur avec un hachage autorisé est trouvé, le classeur s'ouvre.

Pour autoriser une clé USB, branchez-la, sélectionnez-la dans la liste des lecteurs dans XLS Padlock, puis cliquez sur "Allow this USB disk". Vous pouvez autoriser plusieurs clés USB.

REMARQUE

La protection par clé USB offre moins de flexibilité et de sécurité que la protection par dongle dédié.

Désactivation

Introduction à la désactivation

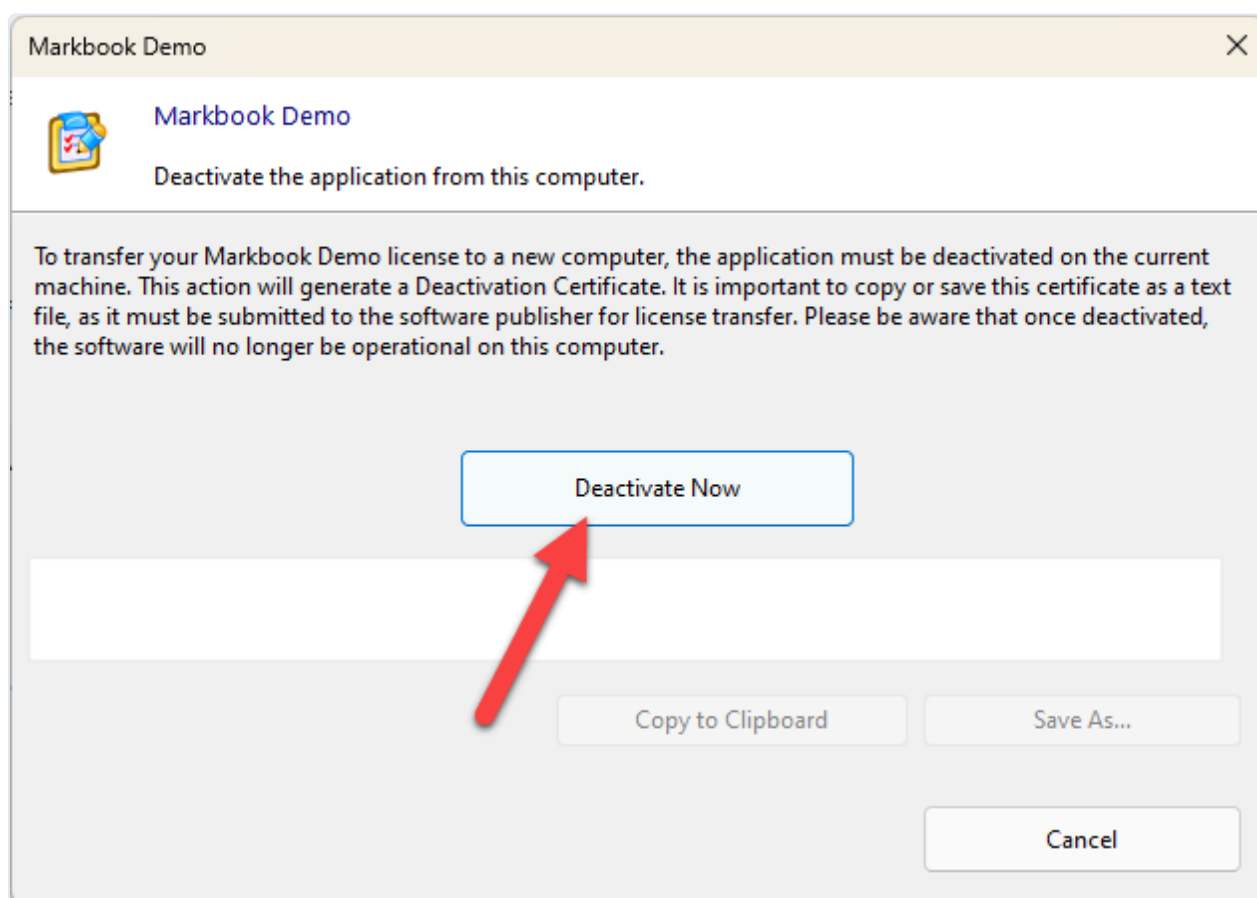
XLS Padlock fournit un système de désactivation robuste qui permet aux utilisateurs finaux de désenregistrer leur application d'un ordinateur. Ceci est particulièrement utile lorsque les clients doivent transférer une licence vers une nouvelle machine ou réactiver leur logiciel après des modifications importantes du système.

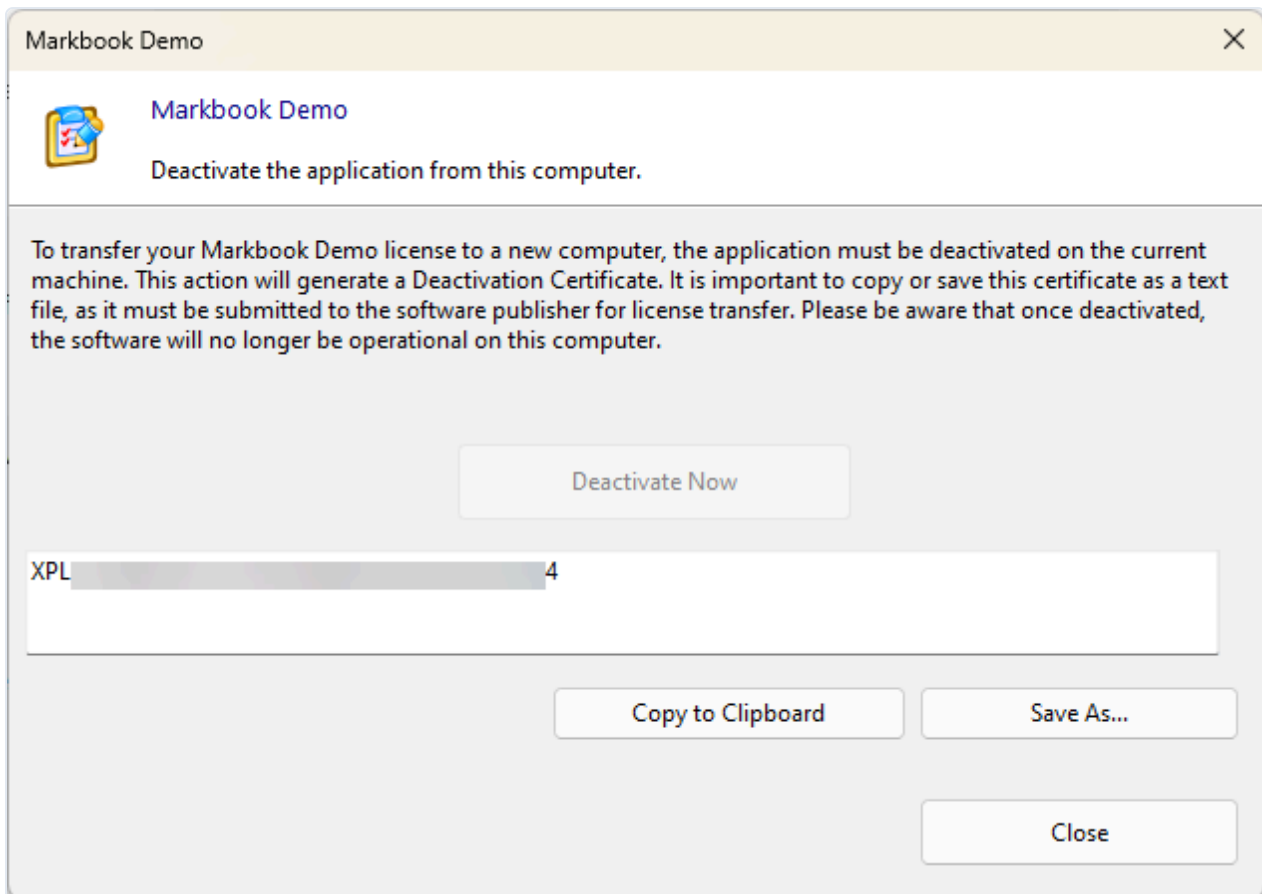
Comment fonctionne la désactivation

La désactivation génère un certificat que l'utilisateur final doit vous envoyer. Grâce à la fonction **Test Deactivation Certificate** de XLS Padlock, vous pouvez vérifier l'authenticité du certificat. Il est essentiel de noter qu'une fois une application désactivée, son Activation Key est invalidée. Vous devrez délivrer une nouvelle clé à l'utilisateur pour toute activation future.

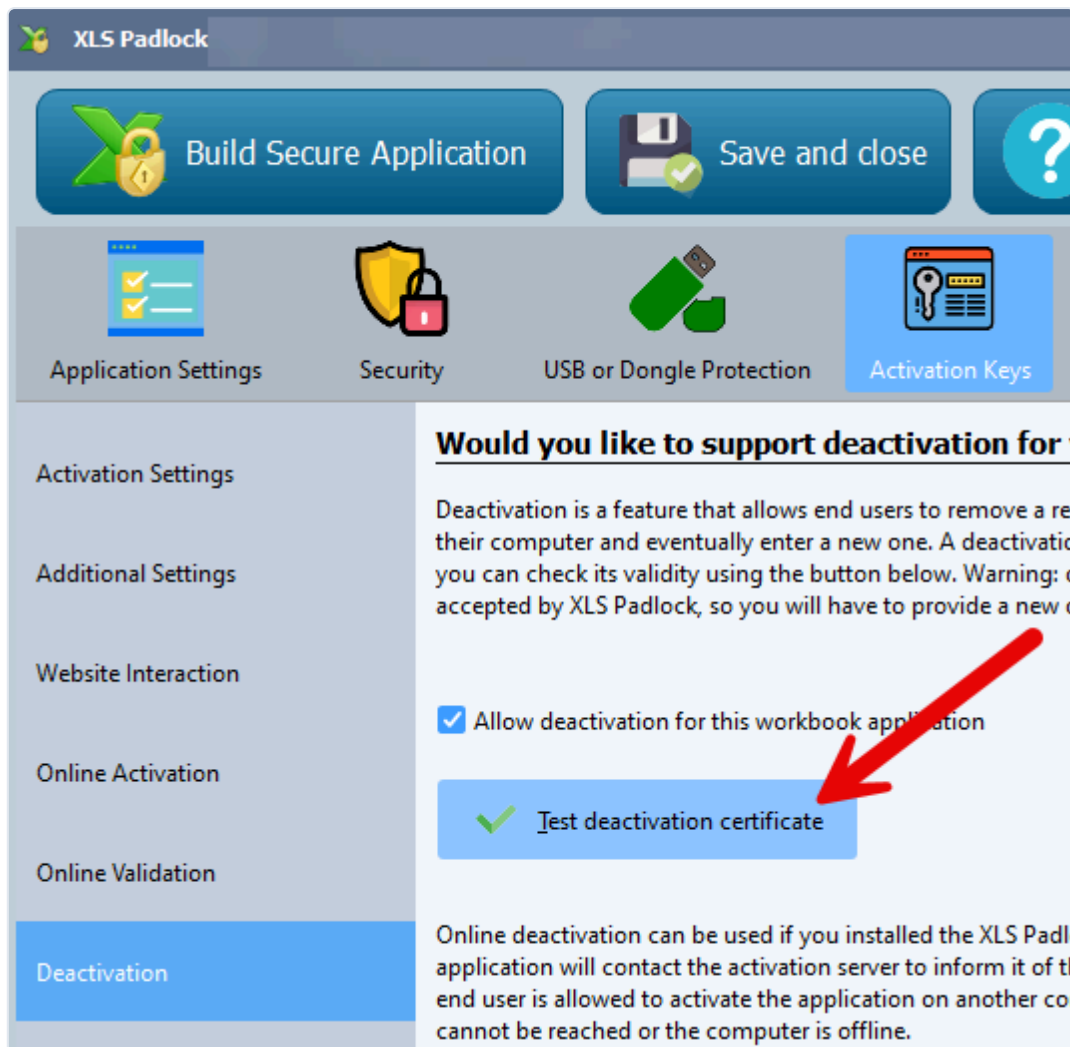
Désactivation manuelle

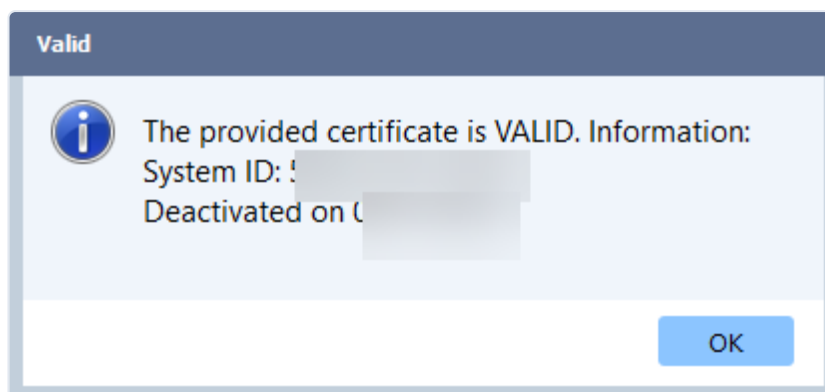
Lorsqu'un utilisateur désactive son application, il doit générer un certificat de désactivation et vous l'envoyer :





À réception, vous pouvez utiliser le bouton 'Test deactivation certificate' dans XLS Padlock pour vérifier le certificat. Si la désactivation réussit, XLS Padlock affichera la date de désactivation et l'identifiant unique de l'ordinateur de l'utilisateur :

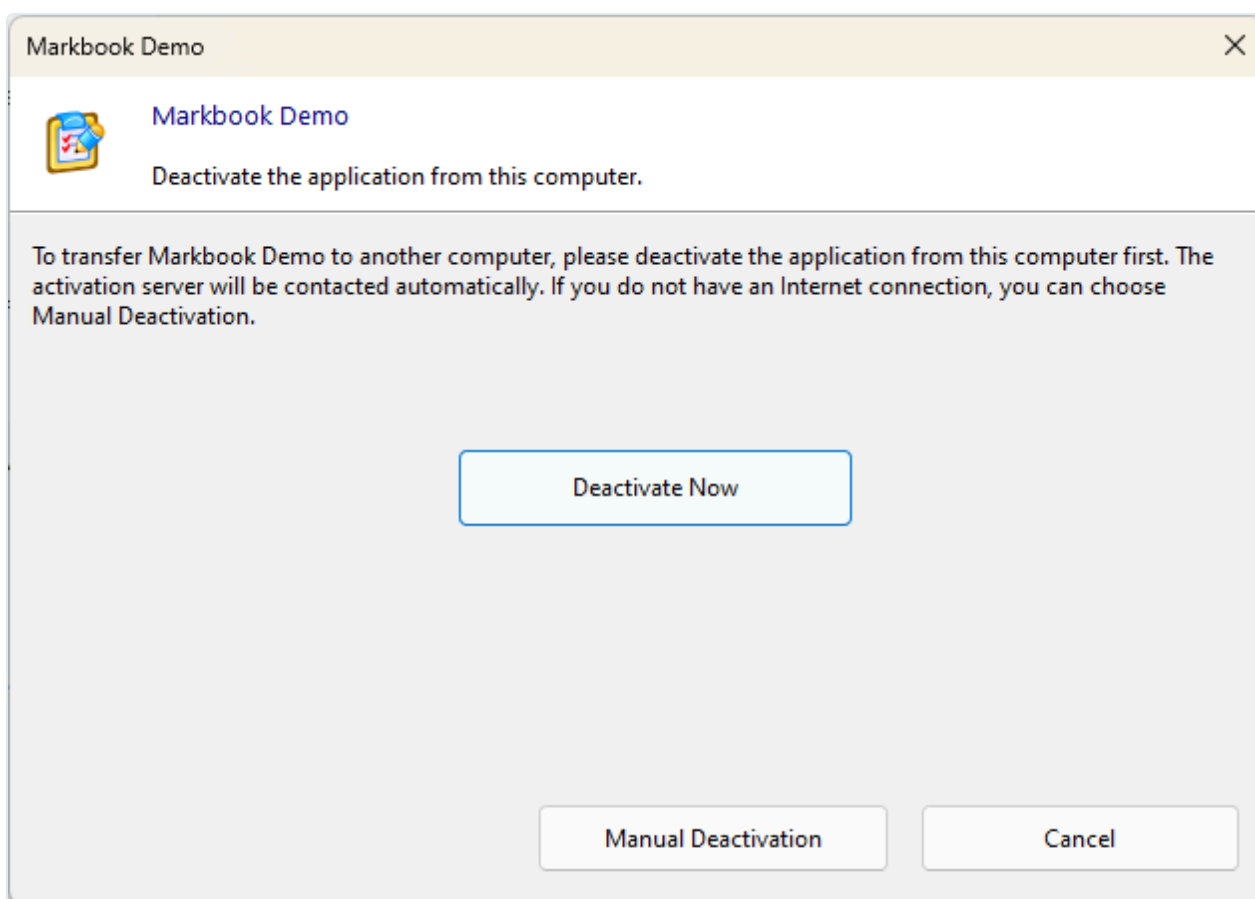




Ces informations vous permettent de gérer les nouvelles activations pour votre client.

Désactivation en ligne

Pour une expérience plus fluide, XLS Padlock prend en charge la désactivation en ligne si vous avez installé le XLS Padlock Activation Kit sur votre serveur web. L'application communique avec votre serveur pour désenregistrer automatiquement l'activation. Si le serveur est inaccessible ou si l'ordinateur est hors ligne, un certificat de désactivation manuelle est créé en solution de repli.



La désactivation en ligne automatise l'ensemble du processus, ce qui le rend plus simple à la fois pour vous et pour vos clients.

Configuration de la désactivation en ligne

Base Deactivation URL

Pour utiliser la désactivation en ligne, vous devez fournir l'URL du XLS Padlock Activation Kit ou du XLS Padlock WooCommerce Integration Kit sur votre serveur web. Par exemple, si vous avez installé l'activation kit dans un sous-dossier nommé « activation », l'URL serait

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

Utilisez HTTPS

Les connexions sécurisées utilisant TLS/SSL sont prises en charge. Vous devriez toujours utiliser des URL commençant par `https://` .

⚠ Si vous préférez gérer la désactivation manuellement, laissez ce champ vide.

Hide Manual Deactivation Button (dans ce cas, il s'affiche si la désactivation automatisée échoue)

Par défaut, la désactivation manuelle est toujours disponible en solution de repli. Si vous souhaitez encourager les utilisateurs à utiliser la désactivation en ligne, vous pouvez choisir de masquer le bouton de désactivation manuelle.

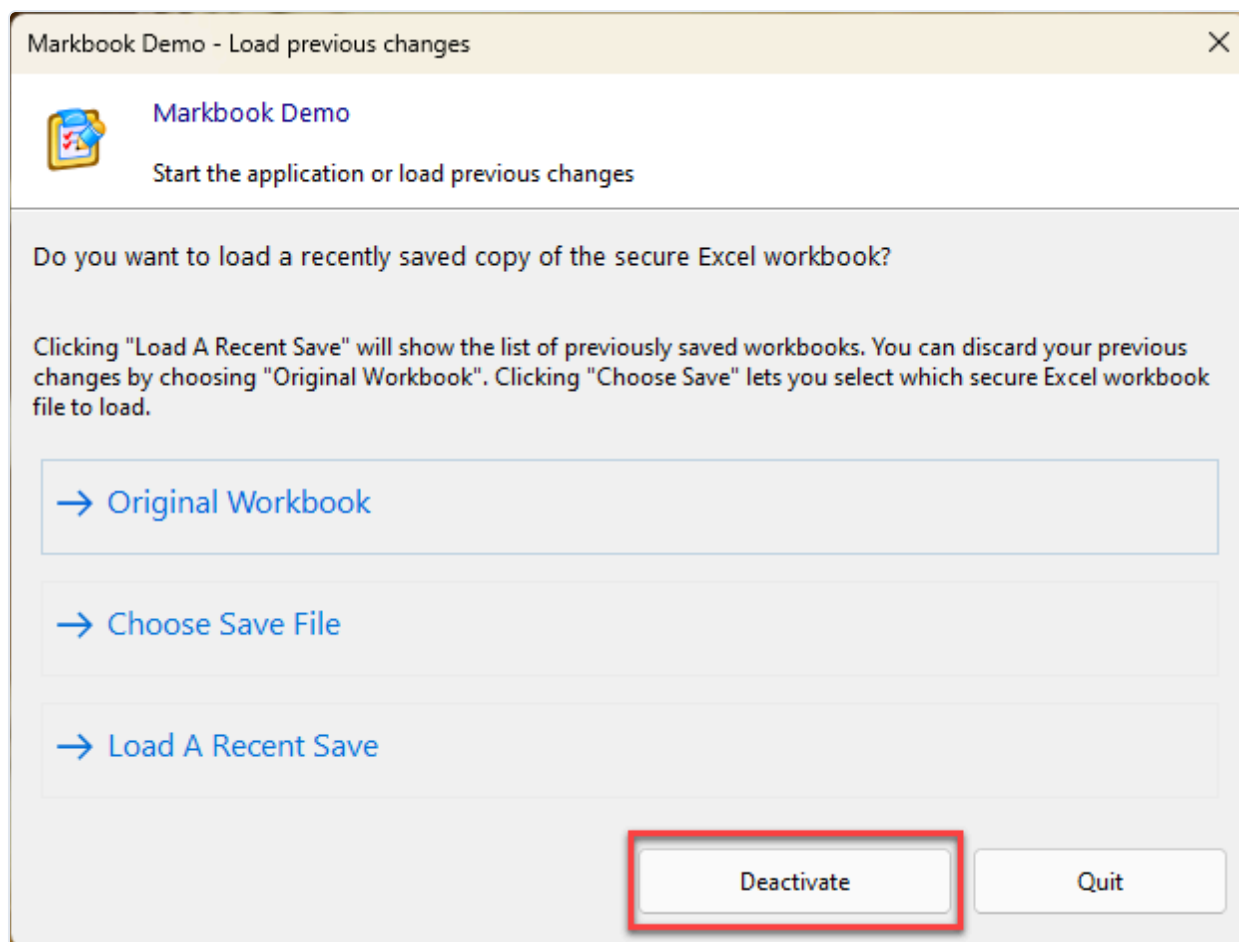
REMARQUE

Même si le bouton est masqué, l'application proposera automatiquement la désactivation manuelle en cas d'échec du processus en ligne (par exemple, si le serveur de désactivation est inaccessible ou si l'utilisateur est hors ligne).

Comment démarrer la désactivation

Un utilisateur peut lancer le processus de désactivation de trois façons :

- La première option passe par la boîte de dialogue Welcome intégrée de l'application. Au lancement de l'application sécurisée, la boîte de dialogue de bienvenue apparaît et présente diverses options, dont un bouton "Deactivate". En cliquant sur ce bouton, les utilisateurs peuvent démarrer le processus de désactivation de manière conviviale et guidée :



- La deuxième méthode fait appel à la ligne de commande. Les utilisateurs avancés peuvent préférer cette méthode pour l'automatisation ou l'intégration système. Pour cela, l'utilisateur doit ouvrir l'invite de commande Windows, se placer dans le répertoire où se trouve le fichier exécutable de l'application sécurisée, puis saisir le nom de l'exécutable suivi du paramètre `**deact**`. Par exemple, si l'exécutable est nommé `Application.exe`, la commande complète serait :

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

En appuyant sur Entrée, le paramètre est transmis au fichier exécutable, ce qui déclenche le processus de désactivation.

- La troisième méthode utilise l'API VBA fournie par XLS Padlock.

La désactivation est définitive sur une machine

Veillez noter qu'une fois une clé désactivée sur un ordinateur, elle est définitivement mise sur liste noire sur cette machine et ne peut pas être réutilisée. La désactivation est une action définitive pour ce système précis. Pour utiliser à nouveau l'application sur le même ordinateur, une nouvelle Activation Key sera nécessaire. Notez que ceci concerne les Activation Keys, et non les jetons d'activation (tels que définis dans le WooCommerce Integration Kit).

👉 Voir aussi : [Activation en ligne](#chapter-online-activation), La boîte de dialogue Welcome intégrée

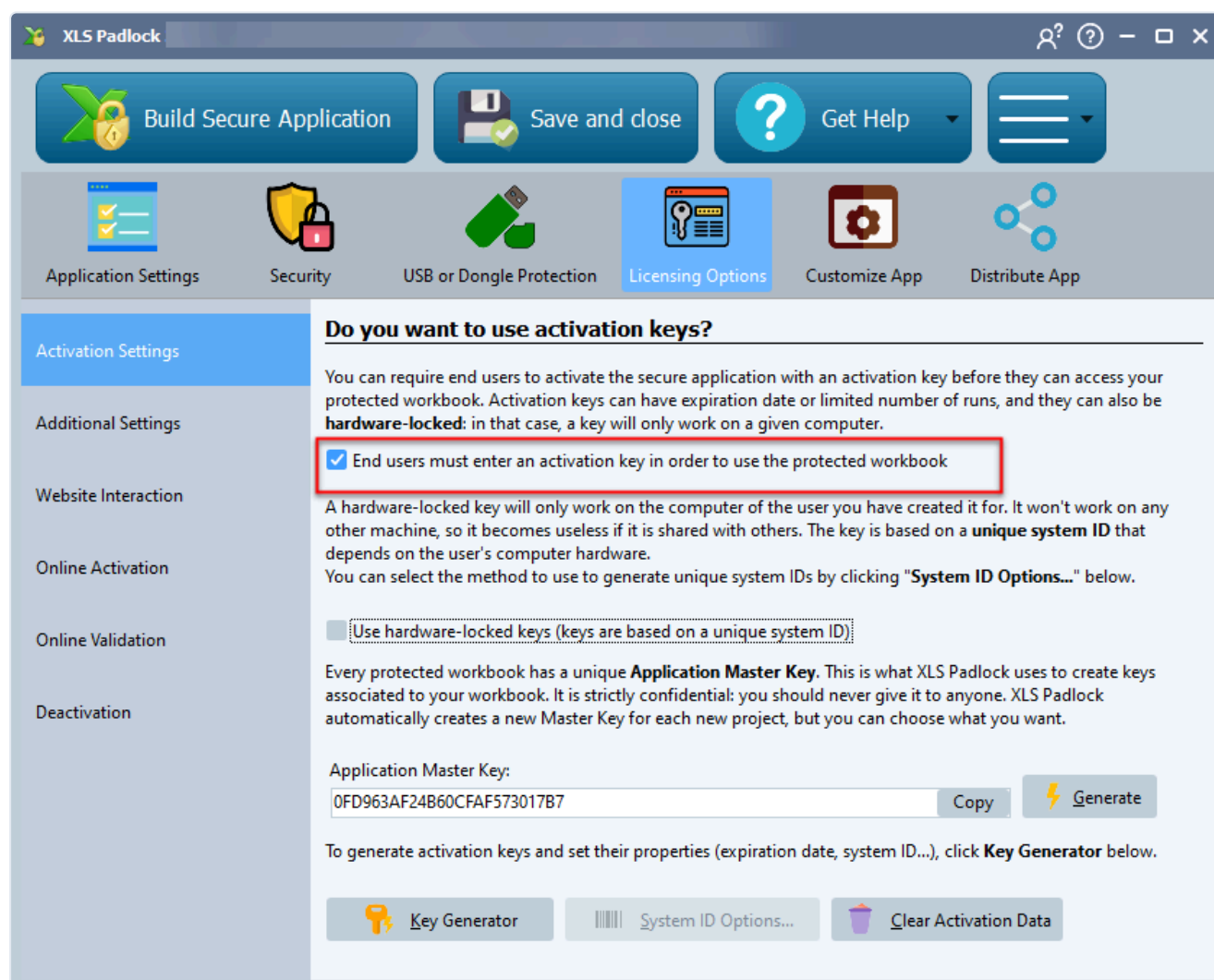
Créer des classeurs d'essai

Vous pouvez offrir un aperçu de vos classeurs Excel à des clients potentiels en créant une version d'essai qui nécessite une licence pour débloquer toutes les fonctionnalités. C'est possible grâce aux [clés d'activation de XLS Padlock](#).

Ce guide vous montre comment configurer une période d'essai pour votre application de classeur.

1. Activer les clés d'activation

Tout d'abord, activez l'option "**End users must enter an activation key in order to use the protected workbook**" (Les utilisateurs finaux doivent saisir une clé d'activation pour utiliser le classeur protégé) et reconstruisez votre application.

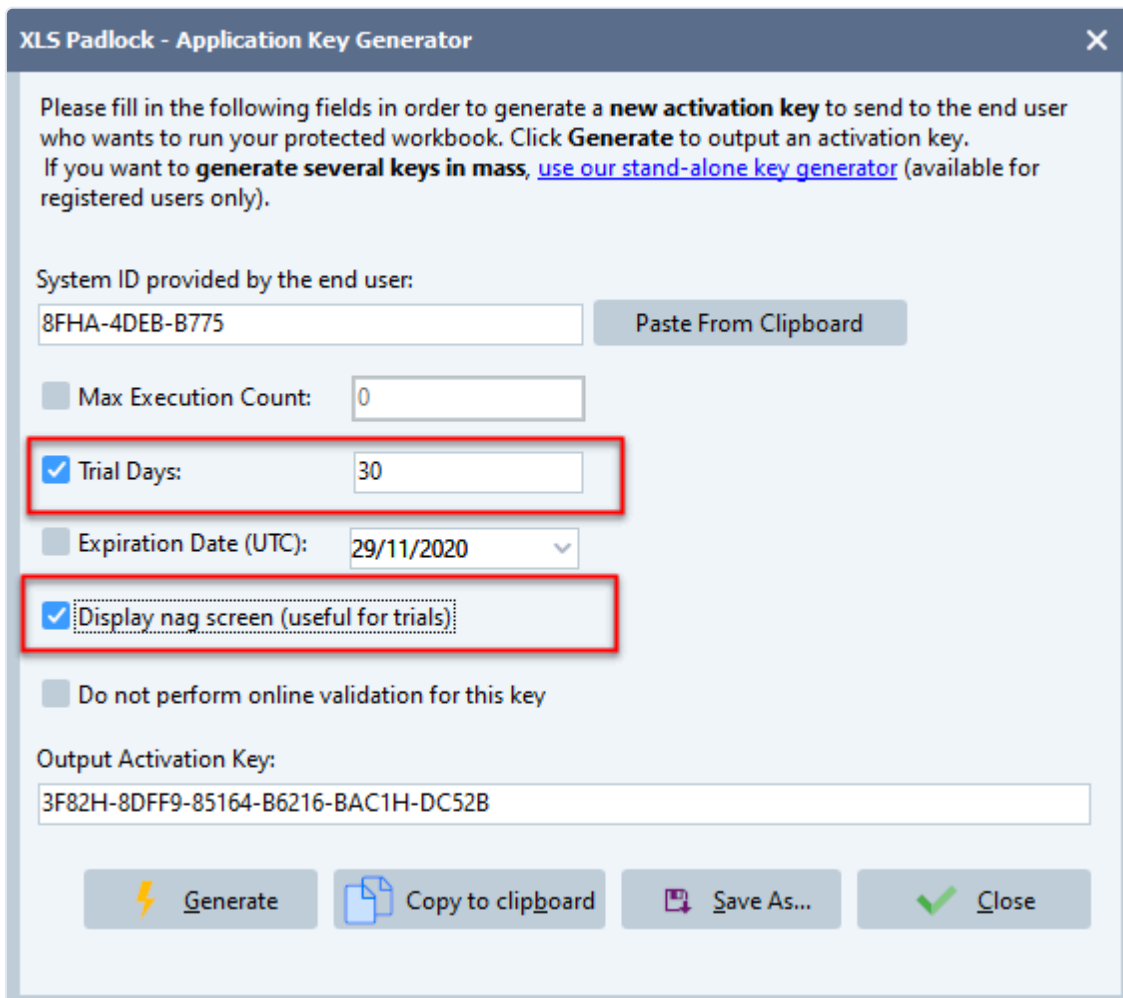


The screenshot shows the XLS Padlock application window. The top toolbar includes buttons for 'Build Secure Application', 'Save and close', 'Get Help', and a menu icon. Below the toolbar is a navigation bar with icons for 'Application Settings', 'Security', 'USB or Dongle Protection', 'Licensing Options' (which is selected), 'Customize App', and 'Distribute App'. The main content area is titled 'Do you want to use activation keys?'. It 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.' Below this text is a checkbox labeled 'End users must enter an activation key in order to use the protected workbook', which is checked and highlighted with a red box. Further down, there is a section for 'Use hardware-locked keys (keys are based on a unique system ID)' with a radio button. Below that, it says '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.' There is a text field for the 'Application Master Key' containing the value '0FD963AF24B60CFAF573017B7', a 'Copy' button, and a 'Generate' button with a lightning bolt icon. At the bottom, there are three buttons: 'Key Generator', 'System ID Options...', and 'Clear Activation Data'.

2. Générer une clé d'activation d'essai

Ouvrez le [Key Generator](#) dans XLS Padlock.

Pour définir une période d'essai, vous pouvez fixer une limite **Trial days** (jours d'essai, par exemple 15 jours), une **Expiration Date** (date d'expiration) fixe, ou un **Max Execution Count** (nombre maximal d'exécutions).



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

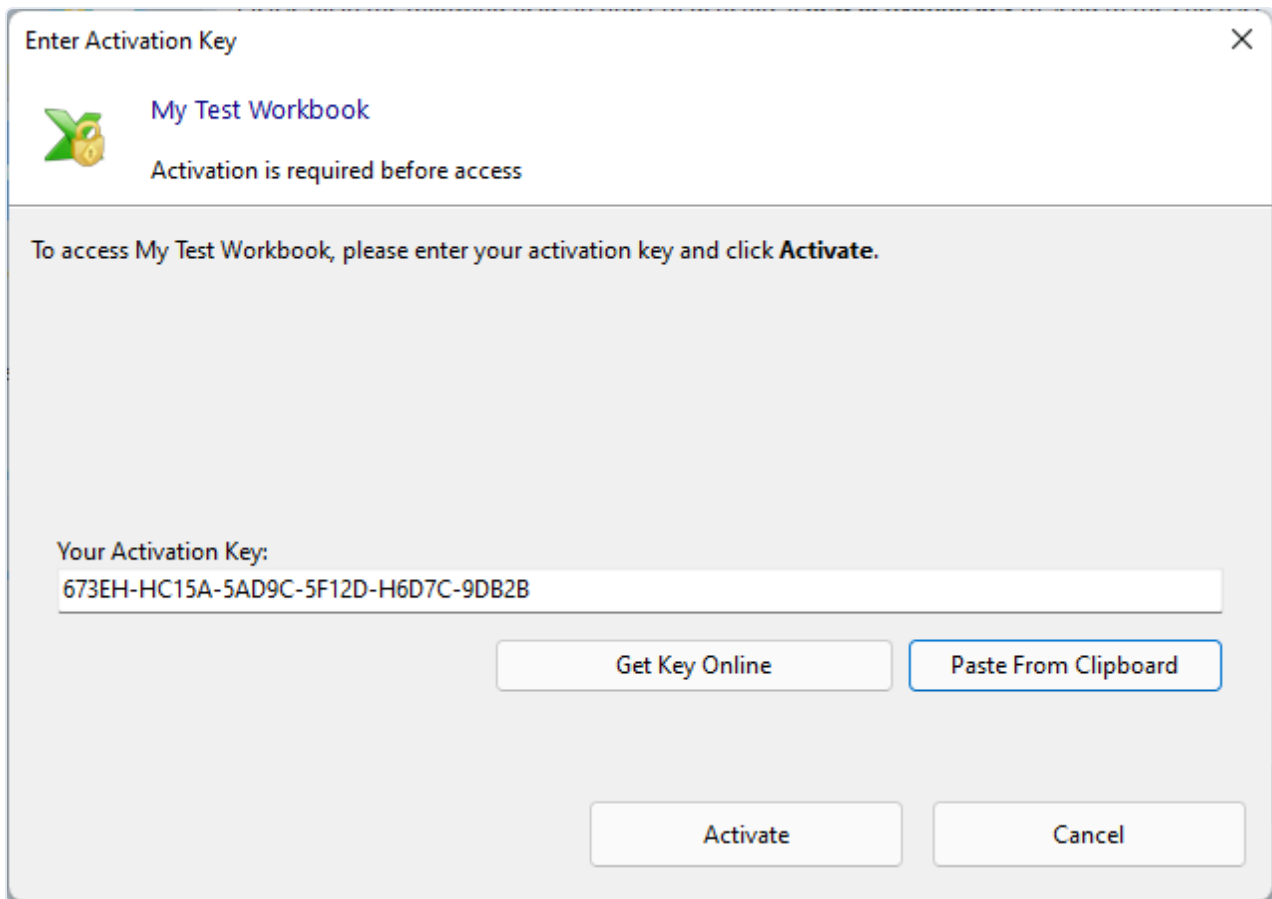
Output Activation Key:
3F82H-8DFF9-85164-B6216-BAC1H-DC52B

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

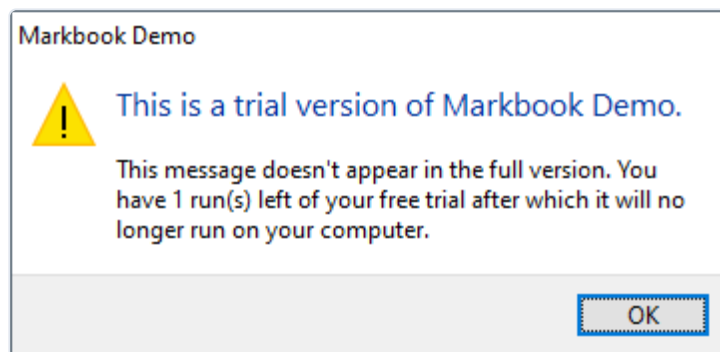
Pour rappeler aux clients qu'ils utilisent une version d'essai, cochez **"Display nag screen"** (afficher l'écran de rappel). Cela affichera une boîte de dialogue de rappel à chaque démarrage de l'application. Cliquez sur **Generate** pour créer la clé d'essai, que vous pourrez ensuite distribuer à vos utilisateurs.

Comment fonctionne l'application en mode essai

Vos clients saisiront la clé d'essai pour activer l'application.



Une fois activé, le classeur s'ouvrira, mais l'écran de rappel apparaîtra pour indiquer la durée d'essai restante.



Lorsque la période d'essai sera terminée, l'application nécessitera une nouvelle clé d'activation non liée à l'essai pour continuer à fonctionner.

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:

Vérifier l'état d'essai avec VBA

Vous pouvez également [vérifier par programmation si le classeur compilé est en état d'essai](#) à l'aide de l'API VBA.

Fermer après un délai donné

Comment puis-je empêcher un utilisateur de laisser une version d'essai ouverte indéfiniment ?

Un utilisateur pourrait obtenir une clé d'essai d'un jour, ouvrir le fichier EXE et le laisser ouvert indéfiniment. L'application continuera à fonctionner. Comment empêcher cela ?

****Solution**** : Vous pouvez utiliser VBA pour configurer une minuterie avec `Application.OnTime` qui ferme automatiquement le classeur après une durée spécifiée. C'est un moyen efficace de faire respecter les limites de l'essai.

Combinez ceci avec les options de sécurité de XLS Padlock pour interdire l'accès à l'éditeur VBA et compiler votre code VBA afin d'empêcher les utilisateurs de désactiver la minuterie.

Mise en œuvre

Insérez le code suivant dans le module `ThisWorkbook` de votre projet 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
```

Comment cela fonctionne

1. **Workbook_Open** : Au démarrage de l'application, cet événement planifie l'exécution de la macro `CloseAndSave` après 8 heures (`TimeValue("08:00:00")`). Vous pouvez ajuster cette durée selon vos besoins (par exemple, `TimeValue("01:00:00")` pour une heure).
2. **CloseAndSave** : Il s'agit de la macro qui effectue l'action. Elle enregistre le classeur, puis le ferme.
3. **Workbook_BeforeClose** : Il s'agit d'une étape de nettoyage essentielle. Si l'utilisateur ferme le classeur manuellement, ce code annule la minuterie en attente, empêchant Excel d'essayer d'exécuter une macro sur un classeur qui n'est plus ouvert, ce qui provoquerait une erreur.

Vérifier l'état d'essai

XLS Padlock fournit une API VBA qui vous permet de déterminer par programmation si votre classeur compilé s'exécute en état d'essai. À l'aide de la fonction ci-dessous, vous pouvez activer ou désactiver des fonctionnalités spécifiques, afficher des messages de mise à niveau, ou créer d'autres limitations d'essai personnalisées en fonction de l'état de la licence de l'utilisateur...

👉 Insérez la fonction suivante dans un module 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
```

Vous pouvez ensuite appeler la fonction :

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

Avertissement

La fonction ne renverra true que si la clé d'activation possède l'indicateur "display nag screen" (essai) :

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:

👉 Voir aussi : [Définir des restrictions sur les clés d'activation](#)

Jours d'essai restants

Cette fonction VBA vous permet de **recupérer par programmation le nombre de jours ou d'exécutions restants** pour un [classeur en version d'essai](#). Elle fonctionne également pour les clés enregistrées qui ont une date d'expiration ou un nombre limité d'exécutions.

👉 Insérez la fonction suivante dans un module 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
```

Vous pouvez ensuite appeler la fonction :

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

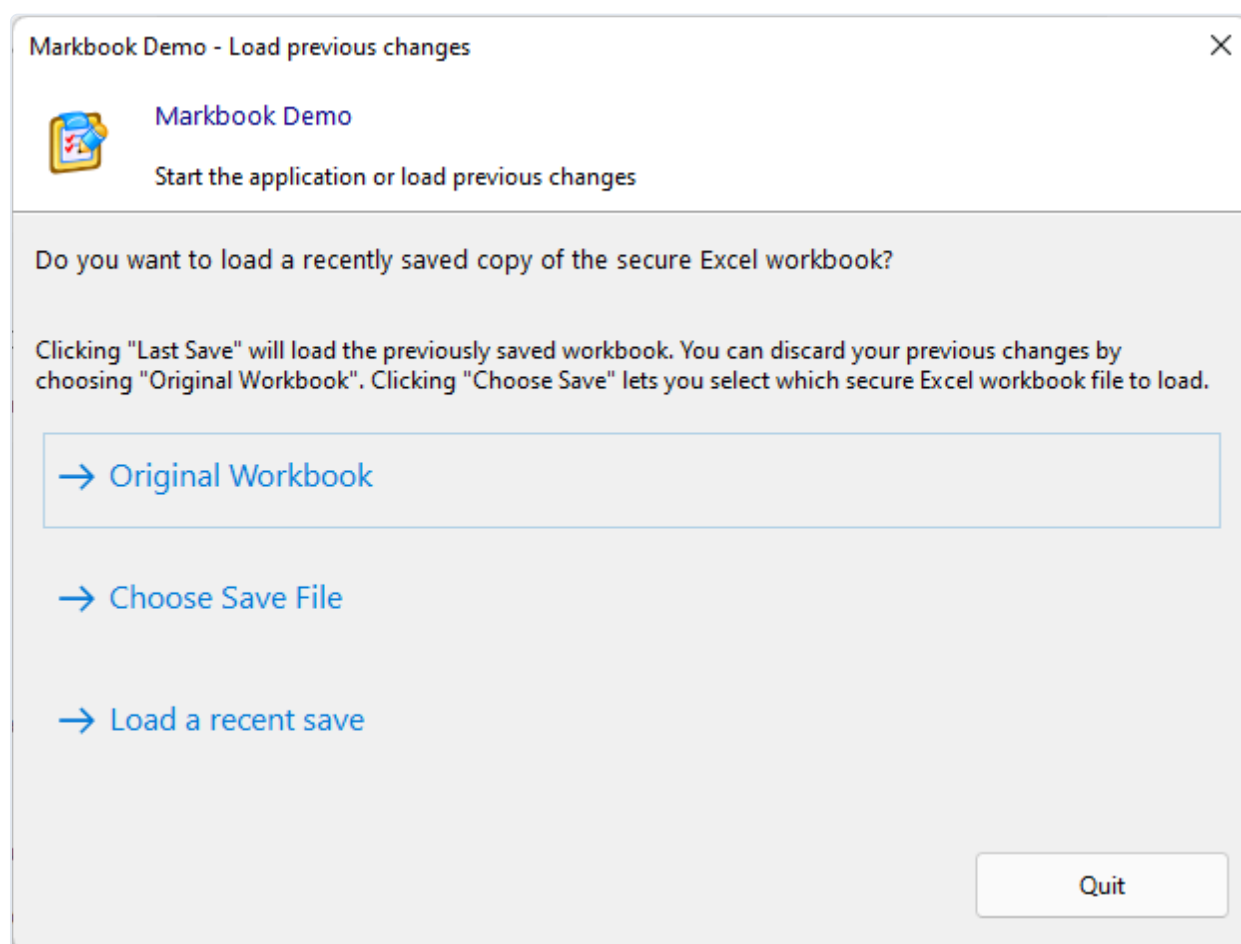
👉 Voir aussi : [Comment créer des classeurs en version d'essai](#)

Options d'enregistrement du classeur

Ces options vous permettent d'affiner la façon dont les modifications du classeur sont enregistrées et chargées lorsque vous [activez l'enregistrement](#).

Always Show the "Load Previous Changes" Welcome Screen

Par défaut, si aucun fichier d'enregistrement n'existe, l'application ouvre directement le classeur d'origine. Lorsque cette option est activée, la boîte de dialogue d'accueil est toujours affichée en premier, ce qui permet aux utilisateurs de commencer avec le classeur d'origine ou de charger un fichier d'enregistrement existant.



Cela est utile si vous distribuez des fichiers d'enregistrement prêts à l'emploi avec votre application ou si vous déplacez l'EXE vers un autre ordinateur et souhaitez charger un fichier d'enregistrement existant.

Save Changes Automatically and Load Them Without Prompt

Cette option fournit un mécanisme d'enregistrement automatique et transparent. Les modifications de l'utilisateur sont enregistrées dans un fichier par défaut lorsque l'application est fermée et sont automatiquement rechargées au prochain démarrage. Aucune boîte de dialogue "Save As" ni aucun écran d'accueil n'est affiché.

Avec ce mode, les utilisateurs ne peuvent pas créer plusieurs fichiers d'enregistrement différents.

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

Cette option augmente la sécurité en verrouillant les fichiers d'enregistrement sur l'ordinateur spécifique où ils ont été créés. Un fichier d'enregistrement créé sur un PC ne peut pas être ouvert sur un autre, ce qui empêche les utilisateurs de partager leurs données enregistrées.

➔ [En savoir plus sur le verrouillage matériel des fichiers d'enregistrement.](#)

Allow Save but Do Not Handle Loading/Saving

Activez cette option si vous souhaitez désactiver les boîtes de dialogue d'enregistrement intégrées de XLS Padlock et gérer vous-même l'ensemble du processus d'enregistrement et de chargement à l'aide de code VBA. Cela vous donne un contrôle total, par exemple pour enregistrer une copie sécurisée sans aucune invite à l'utilisateur.

➔ Découvrez comment [enregistrer une copie sécurisée du classeur sans invite à l'aide de VBA.](#)

Do Not Display the "Original Workbook" Choice

Cette option supprime le choix "Start with the original workbook" de l'écran d'accueil, ce qui oblige les utilisateurs à charger un fichier `.XLSC` ou `.XLSCE` enregistré. Cela est utile si votre application est toujours destinée à être utilisée avec des fichiers de données spécifiques que vous fournissez.

Filet de sécurité automatique

Si le fichier d'enregistrement précédent d'un utilisateur ne parvient pas à se charger, par exemple après que vous avez modifié la [Secret Key](#) pour une nouvelle version, ou changé de mode d'enregistrement entre les versions, le bouton "Original Workbook" est automatiquement réactivé sur l'écran d'accueil pour cette nouvelle tentative. Cela garantit que l'utilisateur peut toujours récupérer l'application au lieu d'être piégé dans une boucle de messages d'erreur.

👉 Voir aussi

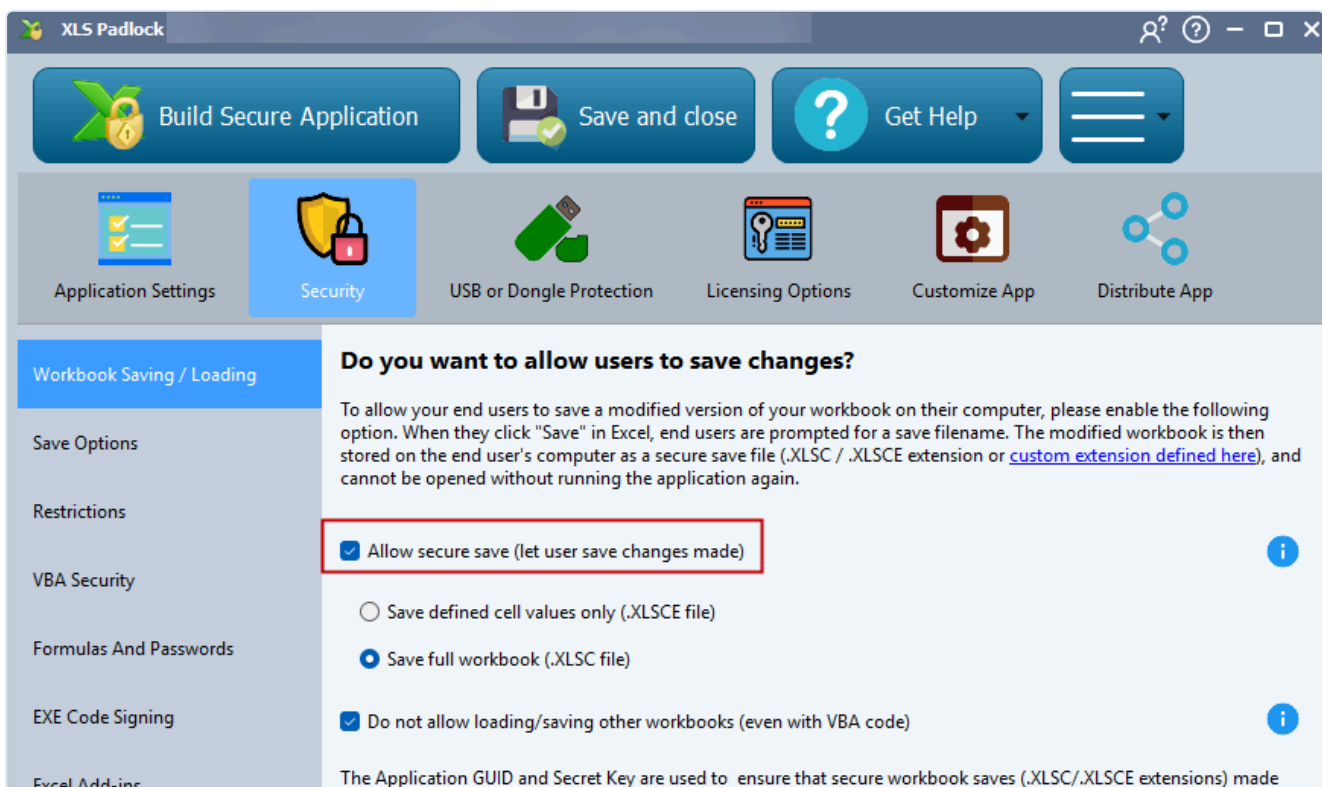
- [Vue d'ensemble de l'enregistrement et du chargement des classeurs](#)
- [Mode d'enregistrement : classeur complet ou valeurs de cellules](#)
- [Recettes VBA pour l'enregistrement des classeurs](#)

Enregistrement et chargement

Avec XLS Padlock, vous avez un contrôle total sur la possibilité pour les utilisateurs d'enregistrer les modifications de votre classeur Excel.

Comment XLS Padlock enregistre et charge les modifications des utilisateurs

Pour activer ou désactiver l'enregistrement, cochez l'option "**Allow secure save**" sur la page Security :

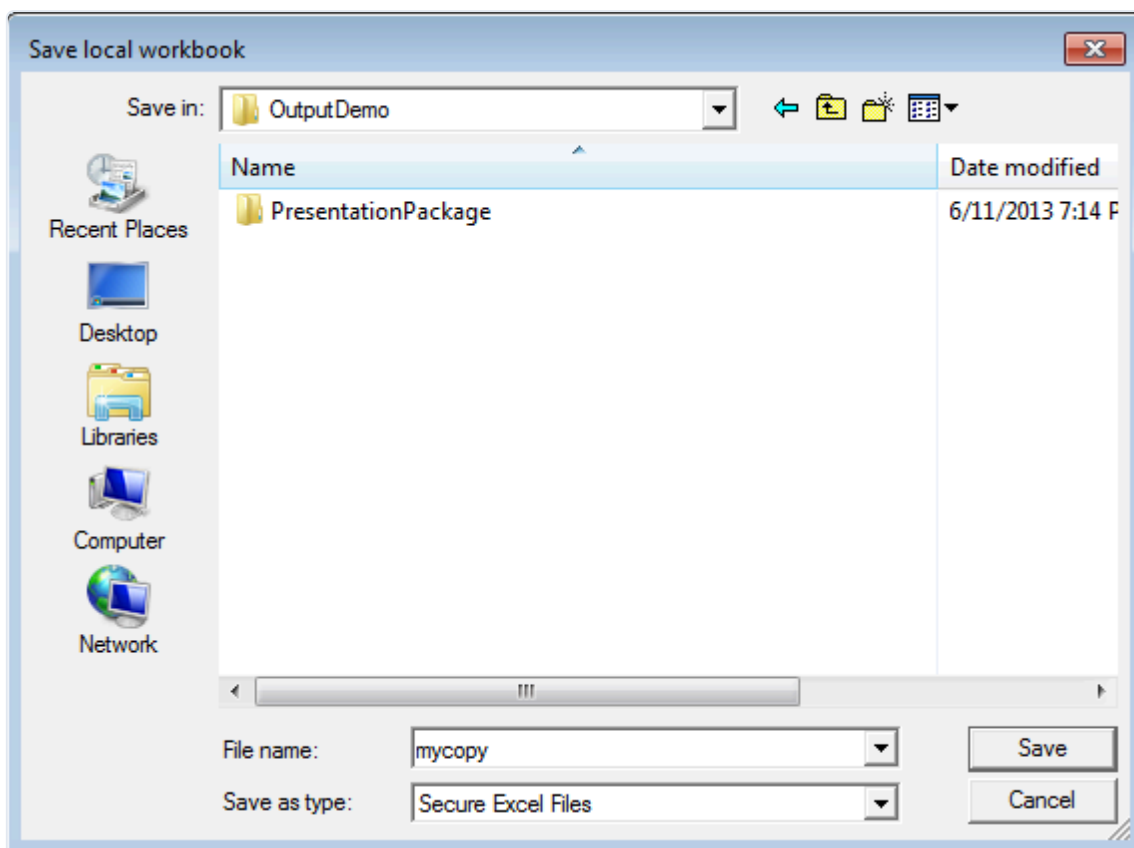


- **Si cette option est désactivée**, votre application ne sera jamais modifiée. Toutes les modifications effectuées par un utilisateur sont supprimées lorsqu'il ferme l'application. Notez que dans certaines versions d'Excel, le bouton Enregistrer n'est pas désactivé et semblera fonctionner, mais aucune modification n'est réellement enregistrée.
- **Si cette option est activée**, vos clients pourront enregistrer leurs modifications. Vous devrez alors [sélectionner un mode d'enregistrement](#).

Pour enregistrer les modifications, les utilisateurs finaux peuvent cliquer sur le bouton Enregistrer standard



dans Excel ou utiliser le menu « Fichier => Enregistrer ». La boîte de dialogue « Enregistrer sous » apparaîtra alors, leur demandant où enregistrer leur copie sécurisée du classeur :



Ce fichier d'enregistrement reçoit l'extension `.XLSC` (ou `.XLSCE`) et **ne peut pas être ouvert sans exécuter à nouveau l'application sécurisée.**

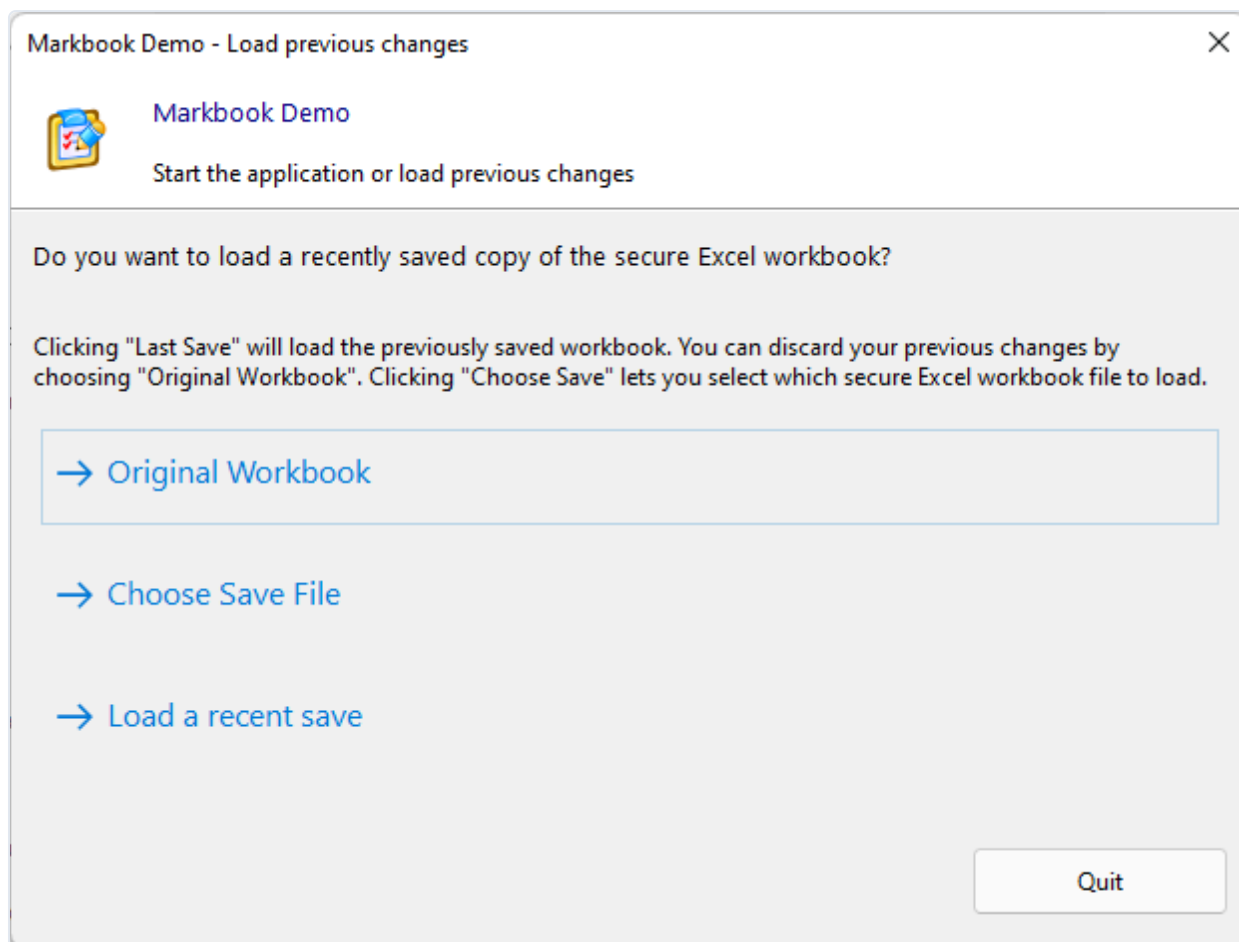
Par défaut, la boîte de dialogue demandera aux utilisateurs finaux s'ils veulent écraser leurs modifications. Vous pouvez désactiver ce comportement dans les [Options avancées](#).

Extension de fichier d'enregistrement personnalisée

Vous pouvez définir votre propre extension personnalisée pour les fichiers d'enregistrement dans les [Options avancées](#).

Comment les utilisateurs finaux chargent leurs modifications

La prochaine fois qu'ils exécuteront votre application, les clients auront le choix d'ouvrir votre classeur d'origine ou une version qu'ils ont précédemment enregistrée. L'écran d'accueil leur permet de "Choose Save File" ou de charger un fichier depuis le menu "Recent Saves".



Par défaut, la boîte de dialogue « Enregistrer sous » s'affiche chaque fois qu'un utilisateur clique sur le bouton Enregistrer. Le dernier fichier enregistré est automatiquement mémorisé. Notez que seule la commande `Save` fonctionne ; `Save As` est toujours désactivée.

Voir aussi

- [Comment migrer les données utilisateur d'une version précédente avec VBA](#)
- [Choisir comment les données sont enregistrées dans le classeur](#)
- [Enregistrer et restaurer par programmation des valeurs personnalisées avec VBA](#)

Mode d'enregistrement : complet ou valeurs de cellules

Si vous autorisez les utilisateurs à enregistrer leur travail, vous devez définir comment XLS Padlock va sauvegarder et charger ces modifications. XLS Padlock propose deux modes de sauvegarde, qui peuvent être sélectionnés sur la page [Workbook Saving and Loading](#).



Sauvegarder le classeur complet (fichier .XLSC)

Dans ce mode, une copie complète et chiffrée du classeur, comprenant toutes les modifications de l'utilisateur, est stockée dans un fichier sécurisé `.XLSC`. Ce fichier ne peut être ouvert que par votre application protégée, ce qui garantit que le classeur reste sécurisé.

Le mode **Full Save** (sauvegarde complète) chiffre et enregistre l'intégralité du classeur exactement tel qu'il est au moment de la sauvegarde. Cela signifie que si vous distribuez ultérieurement un EXE mis à jour, les utilisateurs qui ouvrent un ancien fichier de sauvegarde verront leur travail précédemment enregistré, et non vos nouvelles mises à jour. Leur fichier de sauvegarde est un instantané complet du classeur à ce moment précis.

Les fichiers de sauvegarde complète peuvent être partagés entre utilisateurs, sauf si vous les [verrouillez sur une machine particulière](#). Vous pouvez également [déchiffrer vous-même ces fichiers de sauvegarde](#) pour récupérer les données des utilisateurs.

Quand utiliser le mode de sauvegarde complète

Le mode Full Save est le choix par défaut pour la plupart des utilisateurs et est particulièrement recommandé pour les classeurs complexes dans lesquels les utilisateurs effectuent des modifications importantes sur plusieurs feuilles.

Le principal inconvénient est que les fichiers de sauvegarde des utilisateurs ne sont pas automatiquement compatibles avec les versions mises à jour de votre application. Si vous publiez fréquemment des mises à jour de la logique ou de la conception de votre classeur, envisagez plutôt le mode Cell Values (valeurs de cellules).

👉 Voir aussi : [Comment migrer les données utilisateur depuis une version précédente avec VBA](#)

Sauvegarder uniquement les valeurs de cellules définies (fichier .XLSCE)

Dans ce mode, seules les valeurs de cellules spécifiques et prédéfinies sont enregistrées et restaurées. C'est le choix idéal si vous mettez fréquemment à jour votre classeur source et distribuez de nouvelles versions.

Lorsqu'un utilisateur charge son fichier de sauvegarde `.XLSCE` avec votre nouvel EXE, ses données enregistrées sont chargées dans la structure de votre classeur mis à jour. Cela lui permet de bénéficier de vos mises à jour sans perdre son travail.

Le principal inconvénient est que vous devez manuellement [définir quelles cellules enregistrer](#) avant de compiler votre application.

Comme les fichiers `.XLSC`, les fichiers `.XLSCE` sont chiffrés de manière sécurisée et ne peuvent être ouverts que par votre application. Cela est assuré par une [clé secrète unique](#) propre à votre projet.

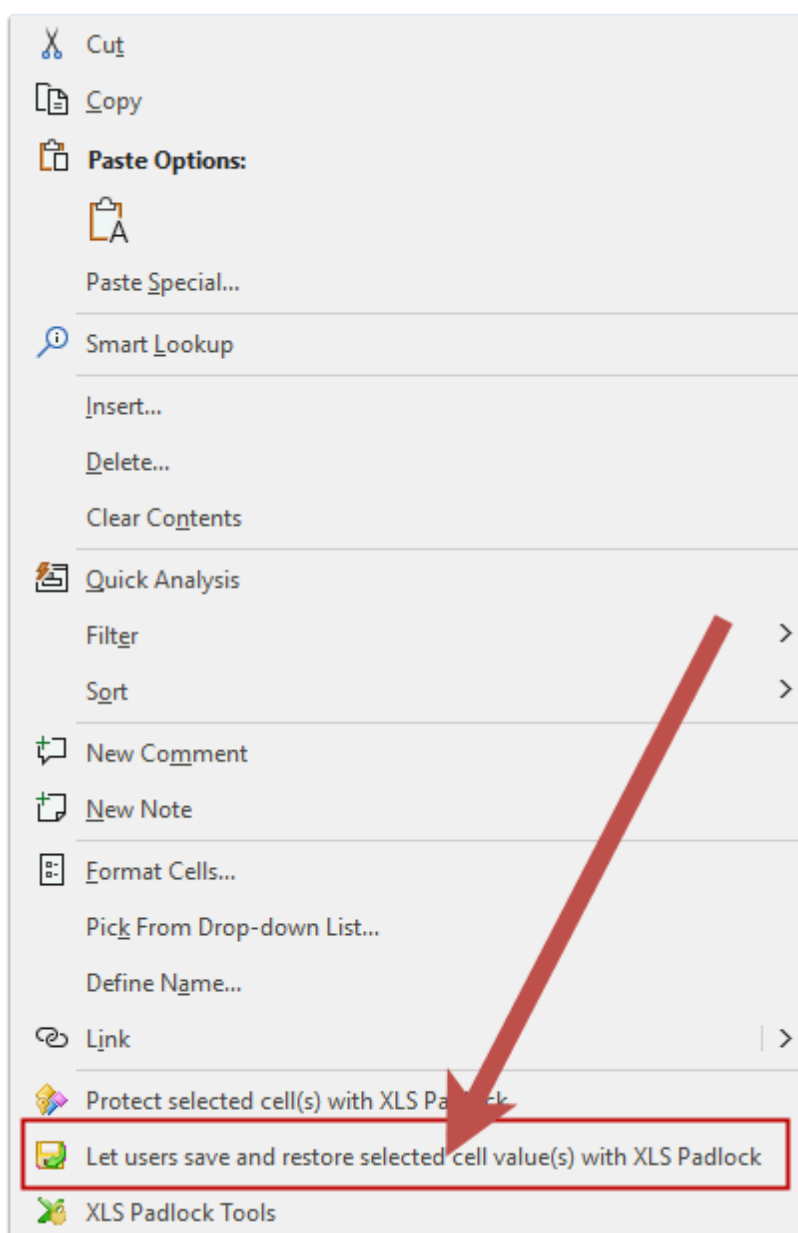
Choisissez votre mode de sauvegarde dès le départ

Les formats Full Save (`.XLSC`) et Cell Values (`.XLSCE`) ne sont pas interchangeables. Une fois que vous avez distribué une application utilisant un mode, passer à l'autre dans une version ultérieure signifie que les fichiers de sauvegarde existants des utilisateurs ne seront pas chargés, XLS Padlock reviendra silencieusement au classeur d'origine intégré pour ces utilisateurs lors du premier lancement de la nouvelle version, et toutes les données qu'ils avaient enregistrées dans le mode précédent sont perdues. Choisissez le mode qui correspond à votre stratégie de mise à jour dès le début du projet et conservez-le.

Définir les cellules à enregistrer et restaurer

Lorsque vous utilisez le mode **Save defined cell values only**, vous devez spécifier quelles cellules votre application va sauvegarder et restaurer.

👉 Pour ce faire, il vous suffit de **cliquer avec le bouton droit sur une ou plusieurs cellules** et de choisir « **Let users save and restore selected cell value(s) with XLS Padlock** » dans le menu contextuel.

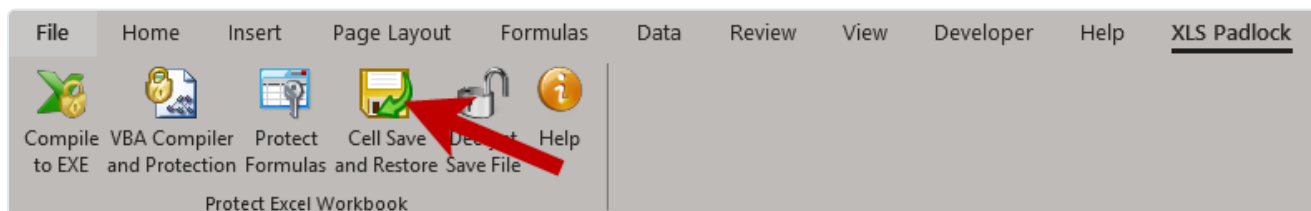


XLS Padlock confirmera que les cellules sélectionnées sont marquées pour la sauvegarde et la restauration.

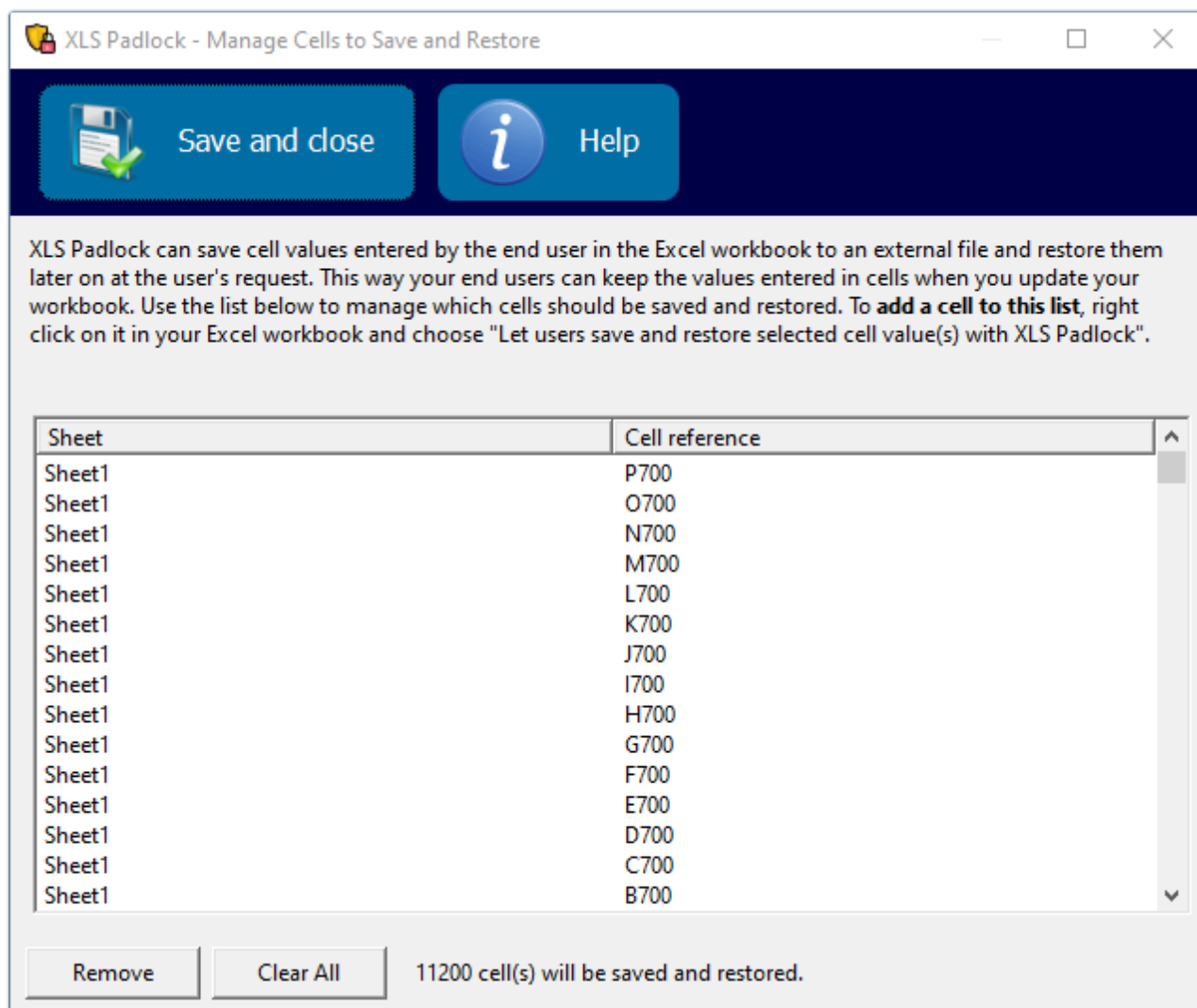
Vous pouvez également [stocker et restaurer des valeurs personnalisées avec du code VBA](#).

Aperçu des cellules à sauvegarder

Pour afficher toutes les cellules configurées pour la sauvegarde, cliquez sur « **Cell Save And Restore** » dans l'onglet du ruban XLS Padlock :



Cela ouvre une fenêtre répertoriant toutes les cellules configurées. À partir de là, vous pouvez supprimer des cellules individuelles ou effacer la liste entière.



Restaurer et enregistrer via VBA

Lorsque vous utilisez le mode **Save defined cell values only**, XLS Padlock vous permet de sauvegarder et restaurer des valeurs personnalisées en plus des valeurs de cellules prédéfinies. Cela est utile pour enregistrer des variables, des paramètres ou des données qui ne sont pas directement stockés dans une cellule.

Aperçu

Cette fonctionnalité repose sur deux événements VBA et deux fonctions de l'API VBA. Vous placez du code dans les sous-routines d'événement pour lire ou écrire vos données personnalisées lorsque l'utilisateur charge ou enregistre son travail.

Événements et fonctions principaux

Événements VBA

Les deux sous-routines d'événement suivantes doivent être placées dans un module de votre classeur Excel. XLS Padlock les appellera automatiquement pendant le processus de sauvegarde et de chargement.

```
' 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
```

Fonctions de l'API VBA

- **WriteCustomCellValue(UniqueID, Value)** : écrit une valeur de chaîne unique associée à un identifiant unique.
- **ReadCustomCellValue(UniqueID, DefaultValue)** : lit une valeur de chaîne unique pour un identifiant donné. Si l'identifiant n'est pas trouvé, la fonction renvoie la `DefaultValue`.

Important

`WriteCustomCellValue` et `ReadCustomCellValue` ne fonctionnent que lorsqu'ils sont appelés depuis les événements `XLSPadlock_SaveCustomValues` et `XLSPadlock_RestoreCustomValues`, respectivement.

Lecture et écriture de valeurs uniques

Voici comment écrire une valeur unique :

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

Et voici comment la relire :

```
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
```

Lecture de toutes les valeurs sous forme de dictionnaire

Vous pouvez également lire toutes les valeurs personnalisées enregistrées d'un coup en passant une chaîne vide comme identifiant à `ReadCustomCellValue`. Cela renvoie un objet `Scripting.Dictionary`.

Référence requise

Pour utiliser cette fonctionnalité, vous devez ajouter une référence à « **Microsoft Scripting Runtime** » dans votre projet VBA (depuis le menu de la VBE, allez dans 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

```

## Exemple complet : sauvegarde/restauration d'une colonne

<div class="aside aside-note">

L'intégralité du code suivant doit être placée dans un seul module.

</div>

Cette fonction d'assistance génère une chaîne séparée par des virgules à partir des valeurs d'u
```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

```

Cet événement est appelé lorsque l'utilisateur enregistre son travail. Il utilise la fonction d'assistance pour convertir l'intégralité de la plage utilisée de la colonne A en une seule chaîne et l'enregistre.

```
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
```

Cet événement est appelé lorsque l'utilisateur charge un fichier de sauvegarde. Il lit la chaîne et restaure les valeurs dans la colonne 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
```

# Accéder au classeur sécurisé et aux fichiers compagnons

---

## REMARQUE

Ce guide explique comment lancer une nouvelle instance Excel depuis votre classeur protégé et lui accorder l'accès au classeur sécurisé ainsi qu'à ses fichiers compagnons.

Par défaut, seule l'instance Excel lancée par votre EXE compilé peut accéder au classeur sécurisé et à ses fichiers associés. Si vous créez une nouvelle instance Excel à l'aide du VBA, elle n'aura pas la permission d'ouvrir ces fichiers protégés.

Pour accorder l'accès, vous devez utiliser l'API XLS Padlock afin de transmettre le handle de fenêtre (HWND) de la nouvelle instance Excel à l'application principale. Cela autorise la nouvelle instance à travailler avec le système de fichiers virtuel.

```
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

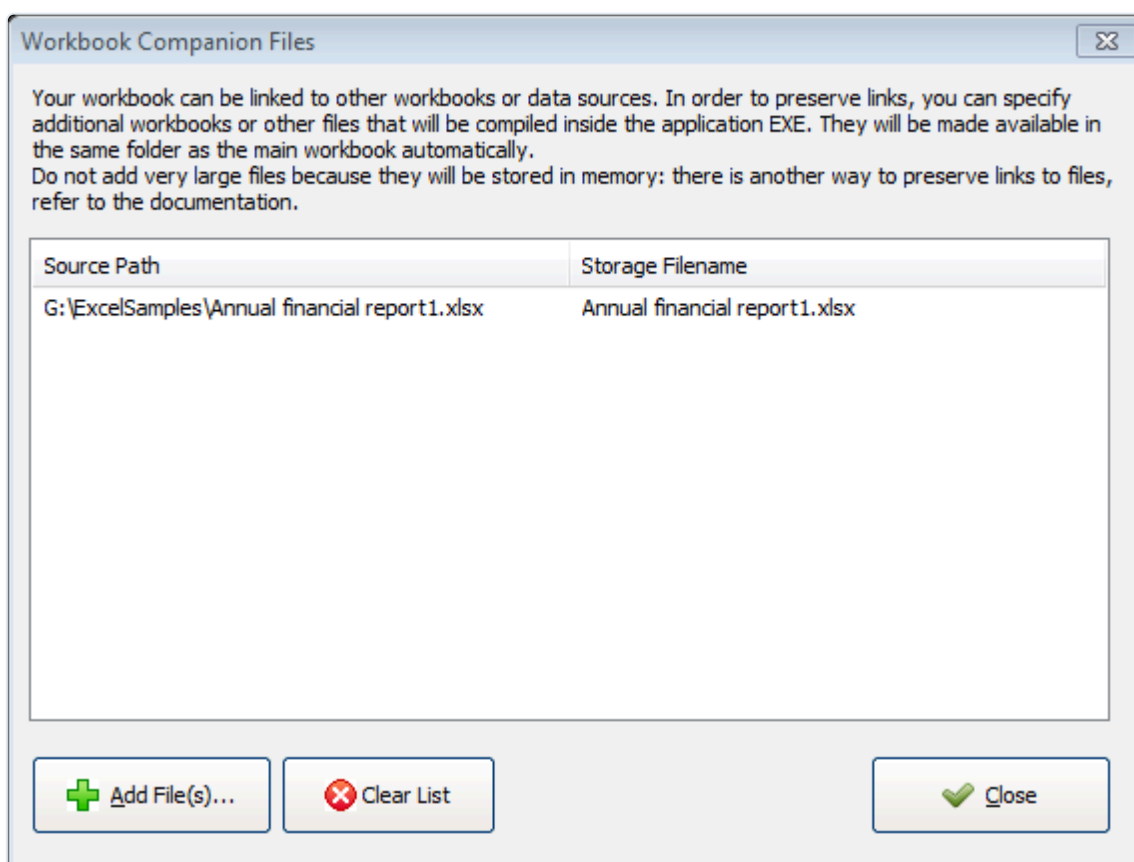
Public Function GetPathToFileInEXEFolder(ByVal Filename As String) As String
' Helper function to get the full path to a file in the EXE's directory.
' See: https://www.xlspadlock.com/doc/get-the-path-to-a-file-in-the-same-folder-as-the-comp
Dim XLSPadlock As Object
On Error Resume Next
Set XLSPadlock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

# Ajouter des fichiers compagnons

Si votre classeur est lié à d'autres classeurs ou sources de données, vous pouvez les ajouter en tant que fichiers compagnons (« companion files »). Cela les regroupe directement dans l'EXE de l'application principale, ce qui facilite la distribution. Lorsque l'application s'exécute, ces fichiers sont rendus disponibles dans le même dossier virtuel que le classeur principal.

Vous pouvez même inclure des compléments XLL.

Par exemple, si votre classeur utilise un fichier texte externe comme source de données, l'ajouter en tant que fichier compagnon garantit qu'il sera toujours trouvé lorsque le classeur compilé est ouvert sur n'importe quel ordinateur.



Pour ajouter des fichiers, cliquez sur **Add Files** et sélectionnez-les. Ils apparaîtront dans la liste. La colonne « Storage Filename » indique le nom de fichier qui sera utilisé lorsque le fichier sera compilé dans l'EXE et restauré à l'exécution.

## Évitez les fichiers volumineux

N'ajoutez pas de fichiers très volumineux, car ils seront entièrement stockés en mémoire. Pour établir un lien vers d'autres classeurs ou fichiers de données volumineux, il est préférable de les placer dans le même dossier que l'EXE de votre application et de les référencer à l'aide de la fonction `PLEvalVar("XLSPATH")`. Consultez [Utiliser les références externes et les hyperliens](#) pour plus de détails.

### ### Accéder aux fichiers compagnons avec le VBA

Pour accéder aux fichiers compagnons avec le VBA, vous pouvez utiliser la fonction suivante :

```
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
```

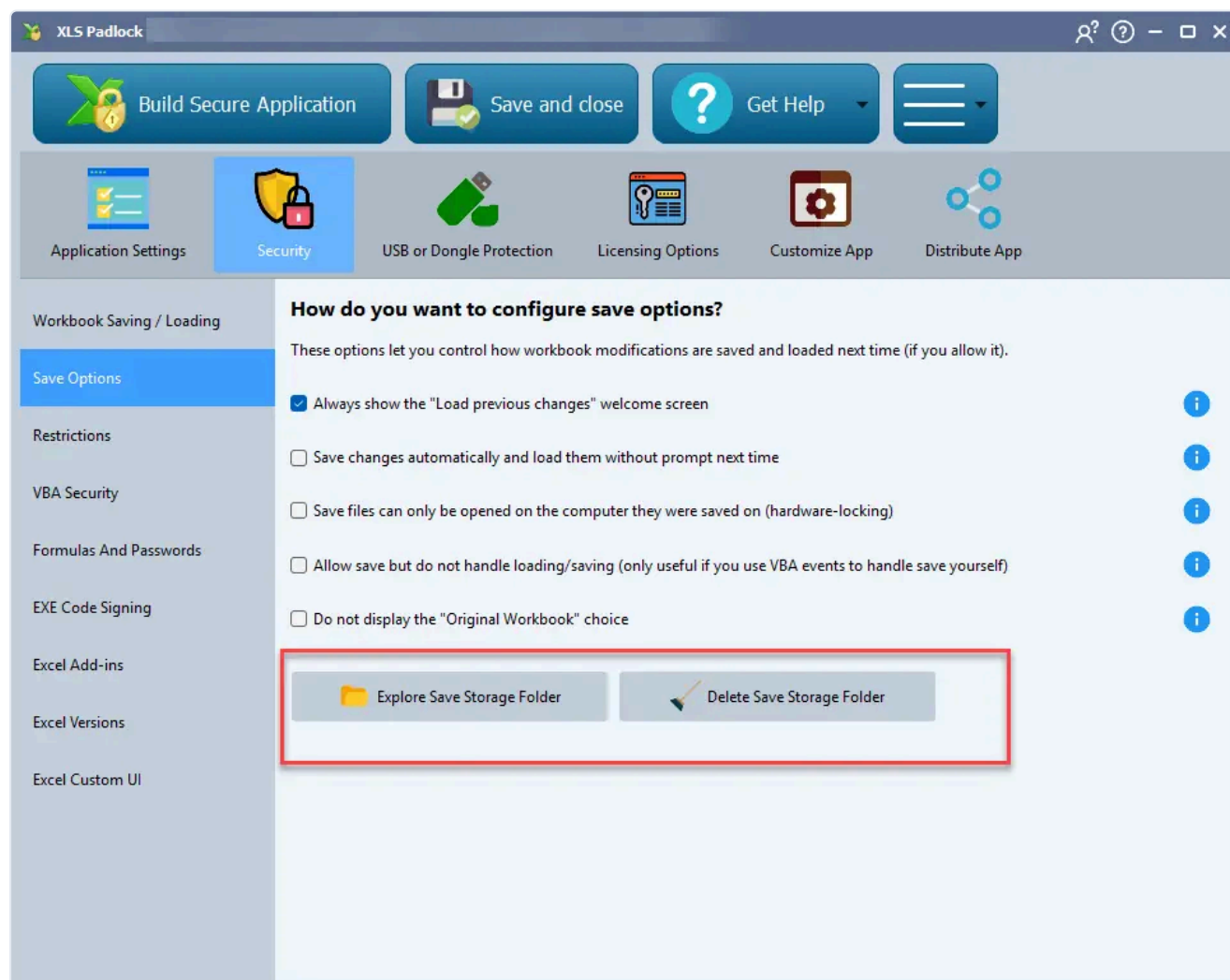
Vous pouvez ensuite utiliser cette fonction pour ouvrir un fichier compagnon :

```
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
```

# Dossier de stockage des enregistrements

Les boutons **Explore Save Storage Folder** et **Delete Save Storage Folder** offrent un accès rapide au dossier local sur votre PC où votre application protégée stocke ses données.



Ce dossier contient : les fichiers enregistrés automatiquement (si cette fonctionnalité est activée), le fichier d'historique des sauvegardes, les données de licence et d'activation.

Si le mode portable est activé (dans les [Options avancées](#)), ces boutons ouvriront le même dossier que celui qui contient le fichier EXE de l'application. Par sécurité, le bouton « Delete » est désactivé dans ce mode.

# Stocker les modifications dans l'EXE

---

Non, les utilisateurs ne peuvent pas enregistrer leurs modifications directement dans le fichier `.exe` compilé. Modifier un fichier exécutable après la compilation peut le corrompre et déclenche souvent des faux positifs des logiciels antivirus.

Pour cette raison, XLS Padlock est conçu pour enregistrer les données utilisateur dans des fichiers de sauvegarde chiffrés distincts (avec les extensions `.xlsc` ou `.xlsce`), et non dans l'EXE principal de l'application.

Considérez votre EXE compilé comme l'application elle-même (comme `Excel.exe`), et les fichiers de sauvegarde comme les documents avec lesquels elle travaille.

# Restreindre le chargement et l'enregistrement

Une vulnérabilité potentielle est que les classeurs Excel peuvent être enregistrés sur le disque à l'aide de commandes VBA ou OLE. Pour empêcher cette méthode d'extraction de données, XLS Padlock inclut une puissante option de sécurité nommée **"Do not allow loading/saving other workbooks"** (Interdire le chargement/enregistrement d'autres classeurs).

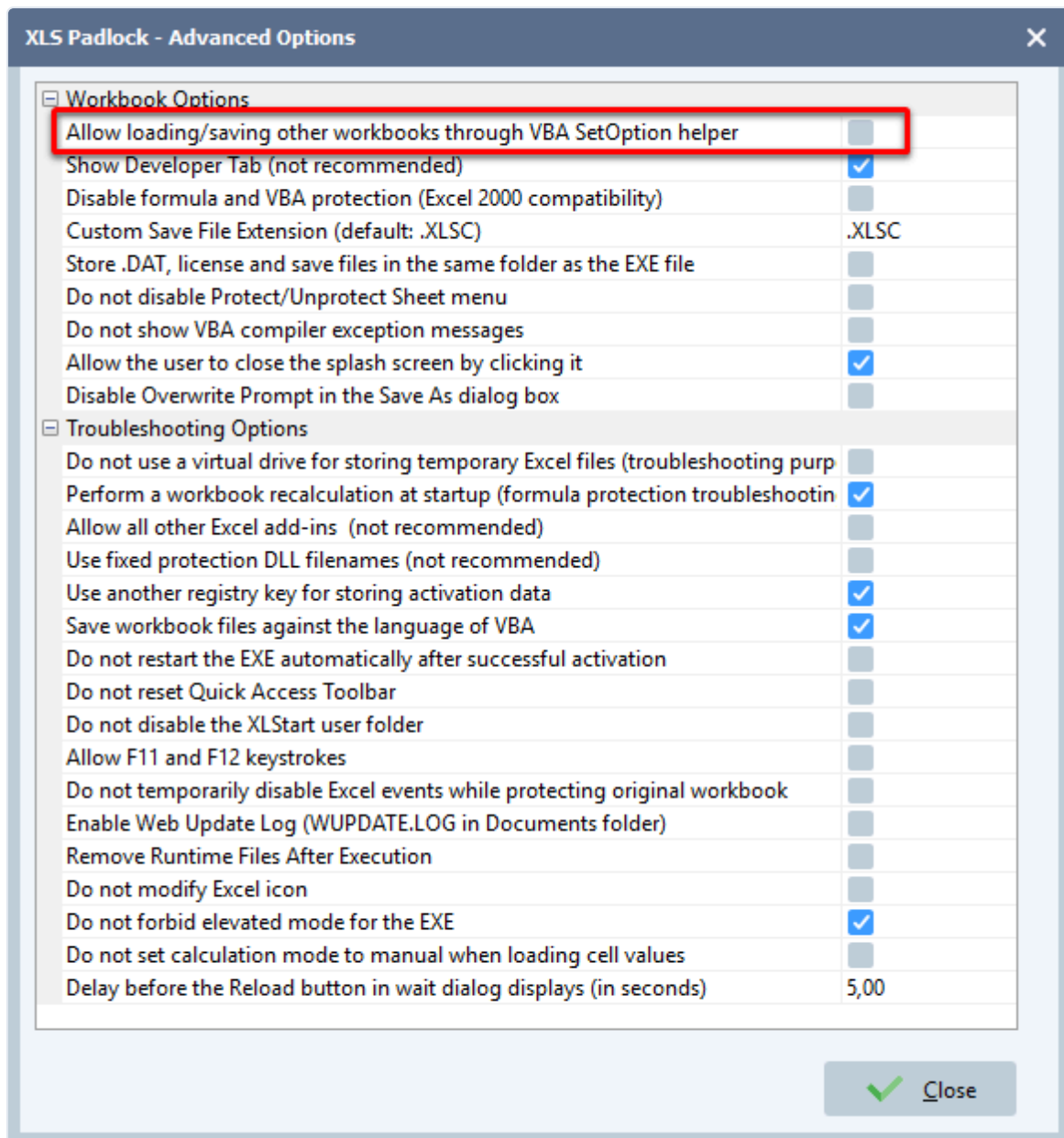
Lorsqu'elle est activée, cette fonctionnalité rend l'instance Excel exécutant votre application **incapable de charger ou d'enregistrer tout autre fichier de classeur**. Cela empêche efficacement les utilisateurs d'ouvrir d'autres classeurs et bloque les hacks basés sur VBA qui tentent de copier des données dans un nouveau fichier non protégé.



## Que faire si mon application doit enregistrer d'autres classeurs ? La solution VBA SetHelper

Cette fonctionnalité de sécurité bloque également le code VBA standard qui enregistre ou charge des classeurs. Si votre application doit légitimement effectuer ces actions, une solution de contournement est disponible.

Vous pouvez désactiver temporairement la restriction dans votre code VBA. Pour ce faire, vous devez d'abord activer l'option avancée **"Allow loading/saving other workbooks through VBA SetOption helper"** ici :



Ensuite, vous pouvez utiliser le code VBA suivant :

```
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

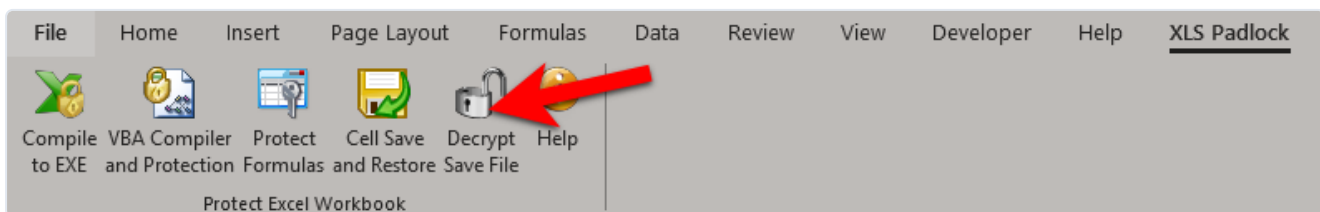
' Your code to save a normal workbook
ActiveWorkbook.SaveAs "D:\My Documents\NormalWorkbook.xlsx"

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

# Ouvrir et déchiffrer les enregistrements

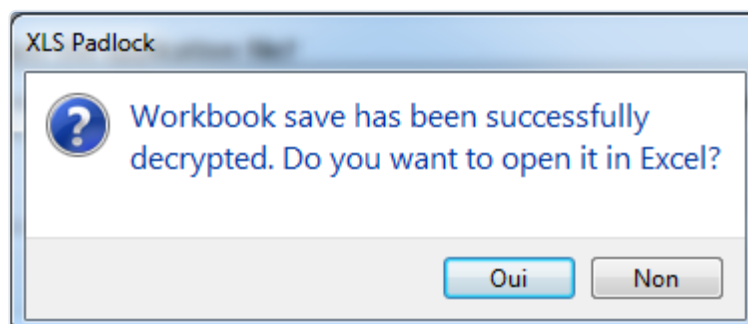
Les fichiers de sauvegarde créés par votre application sont chiffrés et ne peuvent pas être ouverts directement dans Excel. Cependant, en tant qu'auteur original du classeur, **vous pouvez déchiffrer tout fichier de sauvegarde généré par votre application**. Cela est utile pour récupérer des données si un client vous envoie son fichier de sauvegarde.

Pour ouvrir un fichier de sauvegarde ( `.xlsc` ou `.xlsce` ), choisissez "**Decrypt Save File**" dans le ruban XLS Padlock dans Excel :



## Déchiffrement des fichiers `.XLSC` (mode de sauvegarde complète)

Si vous sélectionnez un fichier `.xlsc` créé avec le [mode de sauvegarde complète](#), XLS Padlock déchiffrera l'intégralité du classeur et l'ouvrira.



### Limites du déchiffrement

- Les formules protégées avec XLS Padlock ne peuvent pas être récupérées lors du déchiffrement d'un fichier de sauvegarde. Le classeur déchiffré peut ne pas être entièrement fonctionnel. Cette fonctionnalité ne doit être utilisée que pour récupérer les données saisies par l'utilisateur.
- La même limite s'applique au projet VBA si vous l'avez verrouillé ou compilé.
- Pour utiliser le classeur avec toutes ses fonctionnalités intactes, vous devez exécuter votre application sécurisée et charger le fichier de sauvegarde normalement.

## Déchiffrement des fichiers `.XLSCE` (mode des valeurs de cellules)

Si vous sélectionnez un fichier `.xlsce` créé avec le mode d'enregistrement des valeurs de cellules, XLS Padlock vous demandera si vous souhaitez écraser les cellules de votre classeur actuellement ouvert avec les valeurs du fichier de sauvegarde.

**Cela écrasera vos données**

Confirmer cette opération écrasera définitivement les valeurs des cellules de votre classeur source ouvert.

**Fichier projet requis**

Cette fonctionnalité de déchiffrement nécessite le fichier projet XLS Padlock original ( .xplp ) qui a été utilisé pour créer l'application. XLS Padlock utilise les paramètres de votre projet (comme la Secret Key) pour chiffrer et déchiffrer les fichiers de sauvegarde. Sans le bon fichier projet, vous ne pouvez pas déchiffrer les sauvegardes. **Ne partagez jamais vos fichiers projet XLS Padlock.**

# Verrouiller les enregistrements à une machine

---

Cette option empêche les utilisateurs finaux de partager leurs fichiers de sauvegarde. Lorsqu'elle est activée, l'application crée des **hardware-locked save files** (fichiers de sauvegarde verrouillés matériellement), qui ne peuvent être ouverts que sur l'ordinateur où ils ont été créés. Si un utilisateur tente d'ouvrir un fichier de sauvegarde sur un autre ordinateur, le chargement échouera.

L'application intègre l'**identifiant système** unique de l'ordinateur dans le fichier de sauvegarde et le vérifie au moment du chargement.

## CONSEIL

En tant que propriétaire de l'application, vous pouvez toujours déchiffrer vous-même n'importe quel fichier de sauvegarde verrouillé matériellement à l'aide de la fonction [Decrypt Save File](#) de XLS Padlock.

## Avertissement

Cette fonctionnalité n'est disponible que si vous avez sélectionné le mode Full Save.

# Références externes et hyperliens

XLS Padlock protège un classeur par fichier EXE. Si votre classeur utilise des références externes ou nécessite des fichiers supplémentaires, vous devez mettre à jour vos chemins afin qu'ils fonctionnent correctement après la compilation.

XLS Padlock propose deux moyens principaux pour gérer les fichiers externes :

- Les ajouter en tant que [Companion Files](#).
- Utiliser des chemins dynamiques dans les liens hypertexte ou le code VBA.

Pour gérer les chemins dynamiques, XLS Padlock offre une fonction appelée `PLEvalVar` qui peut être utilisée directement dans les formules Excel ou appelée depuis VBA.

Cette fonction prend un argument de type chaîne de caractères :

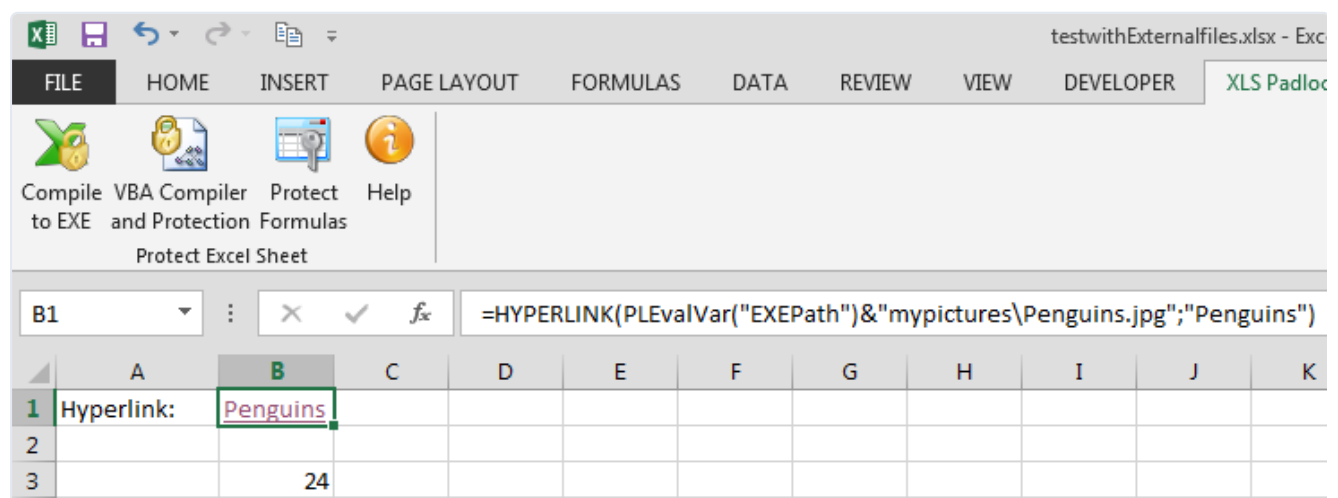
- `=PLEvalVar("EXEPath")` renvoie le chemin complet du dossier qui contient le fichier EXE de l'application (avec la barre oblique inverse finale).
- `=PLEvalVar("XLSPath")` renvoie le chemin complet du dossier qui contient le classeur compilé à l'exécution (avec la barre oblique inverse finale).

## REMARQUE

Ce dossier est un dossier virtuel et, par conséquent, vous ne pouvez pas y placer de fichiers réels. Il n'est utile que si vous travaillez avec des Companion files (voir [Add Companion Files](#)).

## Exemple 1

Vous avez des liens hypertexte vers des fichiers image externes. Ces fichiers image se trouvent dans le même dossier que le fichier XLS du classeur (ou dans un sous-dossier).



Vous avez un lien hypertexte dans une cellule, défini par :

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

Pour le faire fonctionner avec XLS Padlock, vous devez copier tous les fichiers image externes dans le même dossier que le fichier EXE généré avec XLS Padlock. Ensuite, vous devez modifier tous les liens hypertexte afin d'y insérer la fonction `PLEvalVar("EXEPath")` qui renvoie le chemin de ce dossier.

Dans notre cas, cela deviendra :

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

### Avertissement

Les fichiers externes doivent être déployés dans le même dossier que le fichier EXE final de l'application. Il est également recommandé d'éviter les espaces dans les noms de fichiers.

Cela fonctionne aussi pour les fichiers situés dans des sous-dossiers. Un lien comme :

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

...doit être modifié en :

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

## Exemple 2

Pour accéder aux fichiers externes avec VBA, vous pouvez utiliser une fonction d'aide qui construit le chemin complet vers un fichier situé dans le même dossier que l'EXE.

👉 Pour une explication détaillée et un extrait de code réutilisable, consultez le guide expliquant comment obtenir le chemin d'un fichier situé dans le dossier de l'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
```

# Obtenir le chemin à côté du classeur compilé

---

Cette fonction VBA récupère le chemin complet vers un fichier situé dans le même répertoire que l'EXE de votre application. Elle est particulièrement utile pour accéder à des ressources externes ou à des companion files que vous distribuez avec votre classeur protégé.

👉 Insérez la fonction suivante dans un module 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
```

Vous pouvez ensuite appeler la fonction :

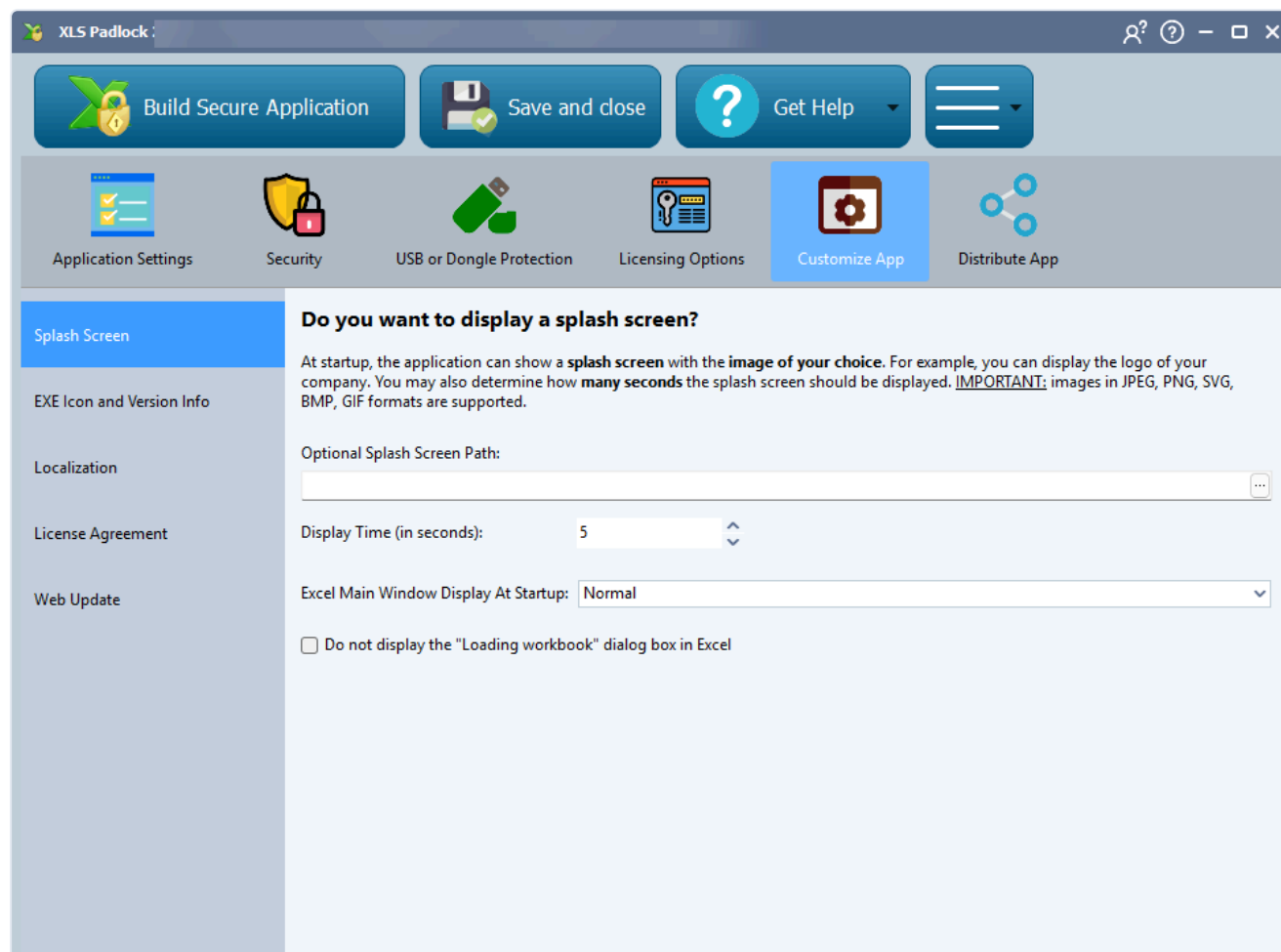
```
Sub Test_File()
 DoSomethingWith(PathToFile("data.xls"))
End Sub
```

👉 Voir aussi : [VBA API Cookbook & Recipes](#)

# Personnaliser l'application

XLS Padlock vous offre plusieurs façons de **personnaliser votre application Excel compilée**, vous permettant d'adapter son apparence, ses messages et son comportement pour les utilisateurs finaux.

Cette rubrique présente un aperçu des options disponibles sous "**Customize App**" dans XLS Padlock :



## Écran de démarrage

Affichez un **écran de démarrage** personnalisé au lancement de votre application, par exemple le logo de votre entreprise ou une image de bienvenue à votre marque.

Vous pouvez choisir le format de l'image (JPEG, PNG, BMP, SVG ou GIF), définir combien de secondes elle reste visible, et contrôler le comportement de la fenêtre Excel pendant le démarrage.

[→ En savoir plus sur les paramètres de l'écran de démarrage](#)

## Icône de l'EXE et informations de version

Remplacez l'icône XLS Padlock par défaut par  **votre propre icône d'application**.

Vous pouvez aussi spécifier des informations de version telles que le **nom du produit**, le **nom de l'entreprise**, le **copyright** et la **description du fichier**, toutes visibles dans la boîte de dialogue des

propriétés de fichier de Windows.

Cela vous aide à donner à votre EXE l'image d'une application autonome professionnelle.

[→ En savoir plus sur l'icône de l'EXE et les informations de version](#)

## Localisation

Personnalisez ou traduisez l'ensemble des **boîtes de dialogue, messages et invites intégrés** affichés par votre classeur protégé.

XLS Padlock fournit un système de localisation flexible afin que vous puissiez adapter votre application à différentes langues ou créer des messages utilisateur entièrement personnalisés.

[→ En savoir plus sur la localisation](#)

## Contrat de licence

Ajoutez un **contrat de licence** que les utilisateurs doivent accepter avant d'exécuter votre application Excel compilée pour la première fois.

Vous pouvez inclure du texte brut ou charger un fichier RTF/HTML externe.

Si les utilisateurs refusent le contrat, l'application ne continuera pas son ouverture.

[→ En savoir plus sur le contrat de licence](#)

## Mise à jour Web

Activez les **mises à jour basées sur le Web** pour votre application Excel.

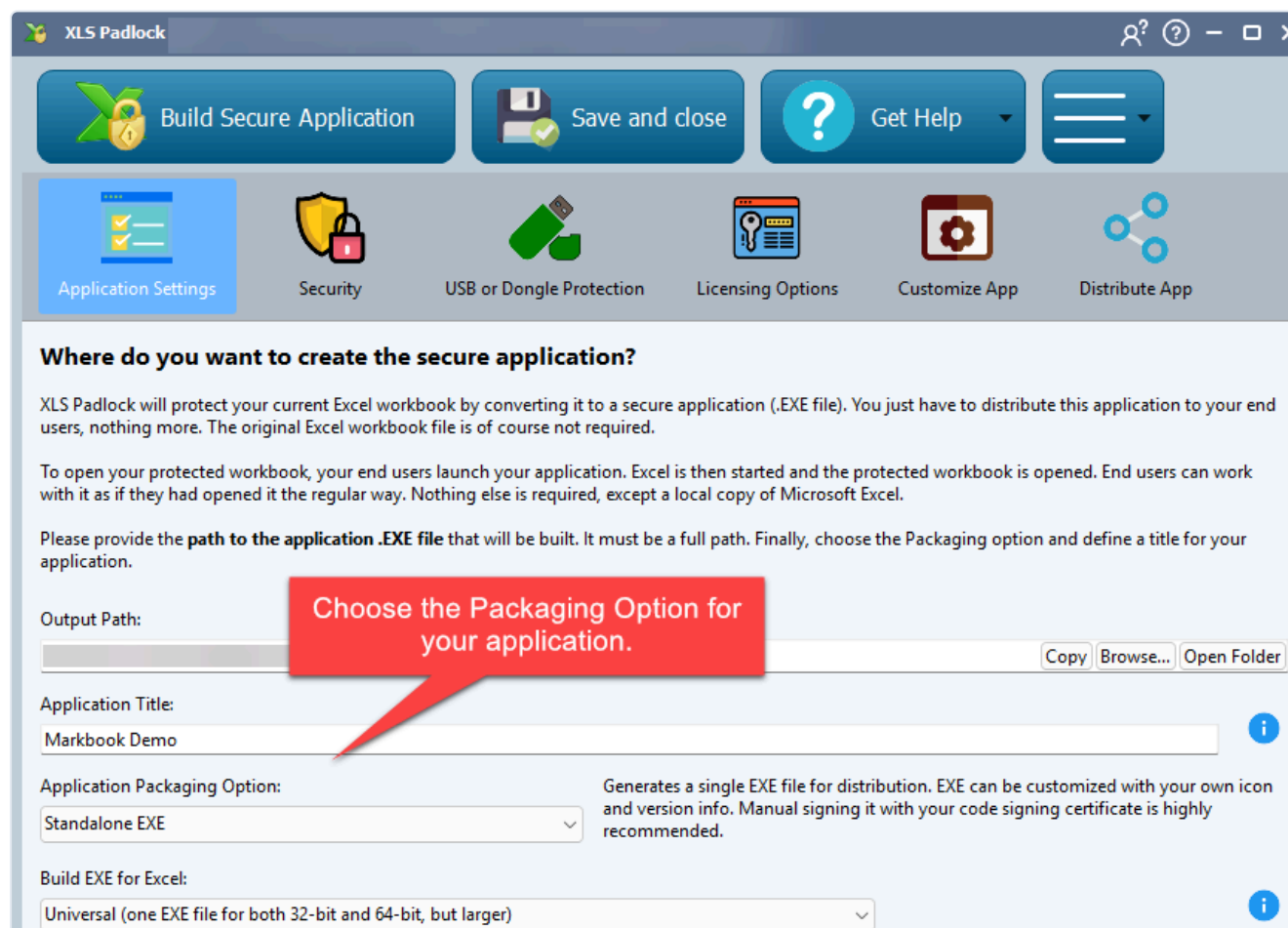
Cette fonctionnalité vous permet d'avertir les utilisateurs lorsqu'une nouvelle version de votre EXE est disponible et de la télécharger automatiquement depuis votre site Web ou votre serveur.

C'est un moyen efficace de garder vos utilisateurs à jour avec un minimum d'effort.

[→ En savoir plus sur la mise à jour Web](#)

# Empaquetage de l'application

L'option **Application Packaging** détermine la façon dont votre classeur sécurisé est compilé, en proposant deux formats distincts.



**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

## Recommandation

Nous recommandons l'option **Standalone EXE** si vous disposez d'un [certificat de signature de code](#). Bien que non obligatoire, la signature de votre EXE est le meilleur moyen d'éviter les avertissements "Unknown App" de Windows SmartScreen et de réduire les faux positifs des logiciels antivirus.

## Standalone EXE

Cette option compile votre classeur Excel en un seul fichier exécutable ( .exe ) autonome. Vous pouvez distribuer ce fichier unique à vos utilisateurs, et ils n'auront pas besoin du classeur Excel d'origine.

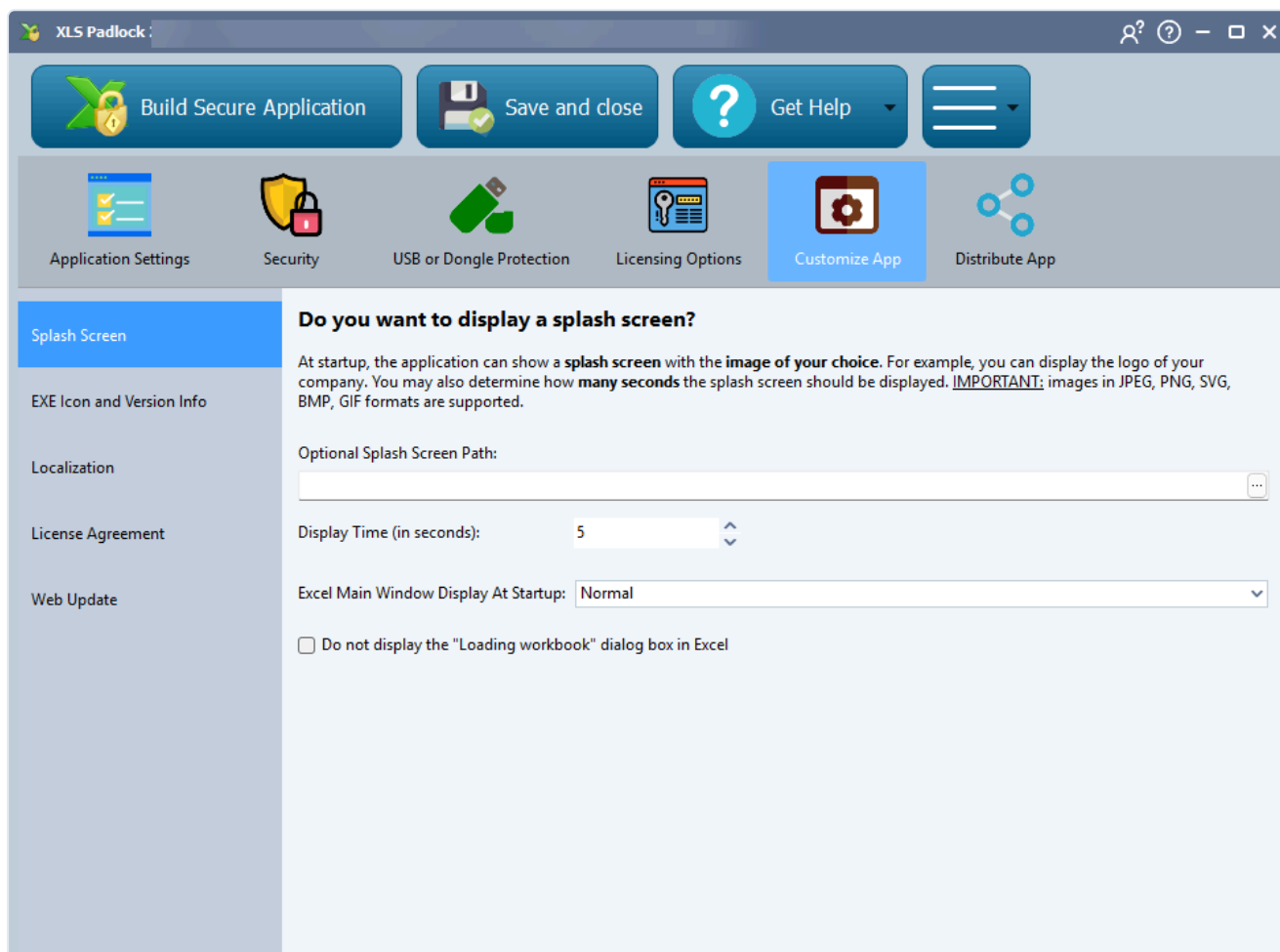
## EXE + XPLAPP Application Bundle

Cette option crée un fichier EXE accompagné d'un fichier compagnon `.bin64` et d'un fichier de données `.xplapp`. Le fichier EXE principal est pré-signé par notre société, ce qui l'aide à être davantage approuvé par Windows SmartScreen et les solutions antivirus.

- Cette méthode de distribution exige que les fichiers EXE, `.bin64` et `.xplapp` soient tous conservés dans le même dossier.
- Pour plus de commodité, XLS Padlock peut regrouper ces fichiers dans une [archive Zip unique ou un installateur](#).
- L'utilisation de cette méthode réduit considérablement les risques de rencontrer des alertes "Unknown Application" et des faux positifs d'antivirus.
- Le fichier compagnon `.bin64` garantit la compatibilité avec les versions 64 bits d'Excel.

# Écran de démarrage

Vous pouvez afficher un **écran de démarrage**, une image montrée brièvement au lancement, pendant que votre application s'initialise. C'est un excellent moyen d'afficher le logo de votre entreprise ou de marquer votre application à votre image.



## Paramètres de l'écran de démarrage

Vous pouvez utiliser n'importe quelle image au format JPEG, PNG, BMP, SVG ou GIF.

Vous pouvez définir la durée pendant laquelle l'écran de démarrage est affiché. Par défaut, les utilisateurs peuvent le fermer immédiatement en cliquant dessus. Ce comportement peut être désactivé dans les [Options avancées](#).

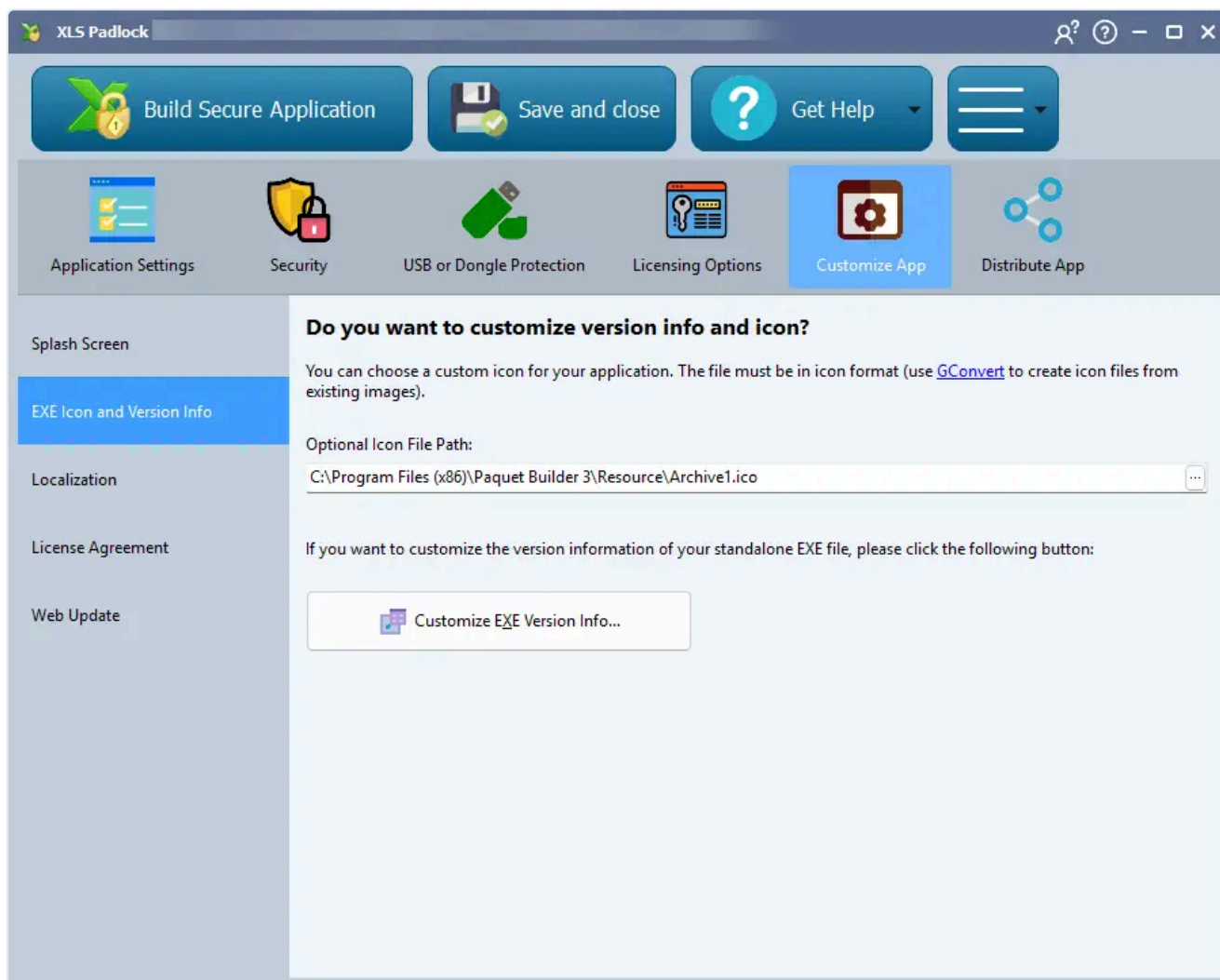
### Utilisez des PNG transparents pour un rendu unique

XLS Padlock prend en charge les écrans de démarrage non rectangulaires et semi-transparentes. Utilisez un fichier PNG 32 bits avec un canal alpha pour donner à votre application une apparence moderne et personnalisée.

👉 Voir aussi : [Ne pas afficher la boîte de dialogue "Loading workbook" dans Excel](#chapter-do-not-display-the-loading-workbook-dialog-box).

# Changer l'icône de l'EXE

XLS Padlock vous permet de remplacer l'**icône d'application par défaut** par la vôtre. Pour personnaliser l'apparence de votre application, indiquez le **chemin vers un fichier d'icône personnalisé**, qui doit avoir une extension `.ico` :



XLS Padlock prend en charge les formats d'icône standard, y compris différentes tailles (par exemple 32x32, 48x48) et profondeurs de couleur. Si vous avez besoin d'un outil pour créer ou extraire des icônes, nous recommandons notre utilitaire compagnon, [GConvert](#), qui peut facilement convertir des images en icônes.

## Avertissement

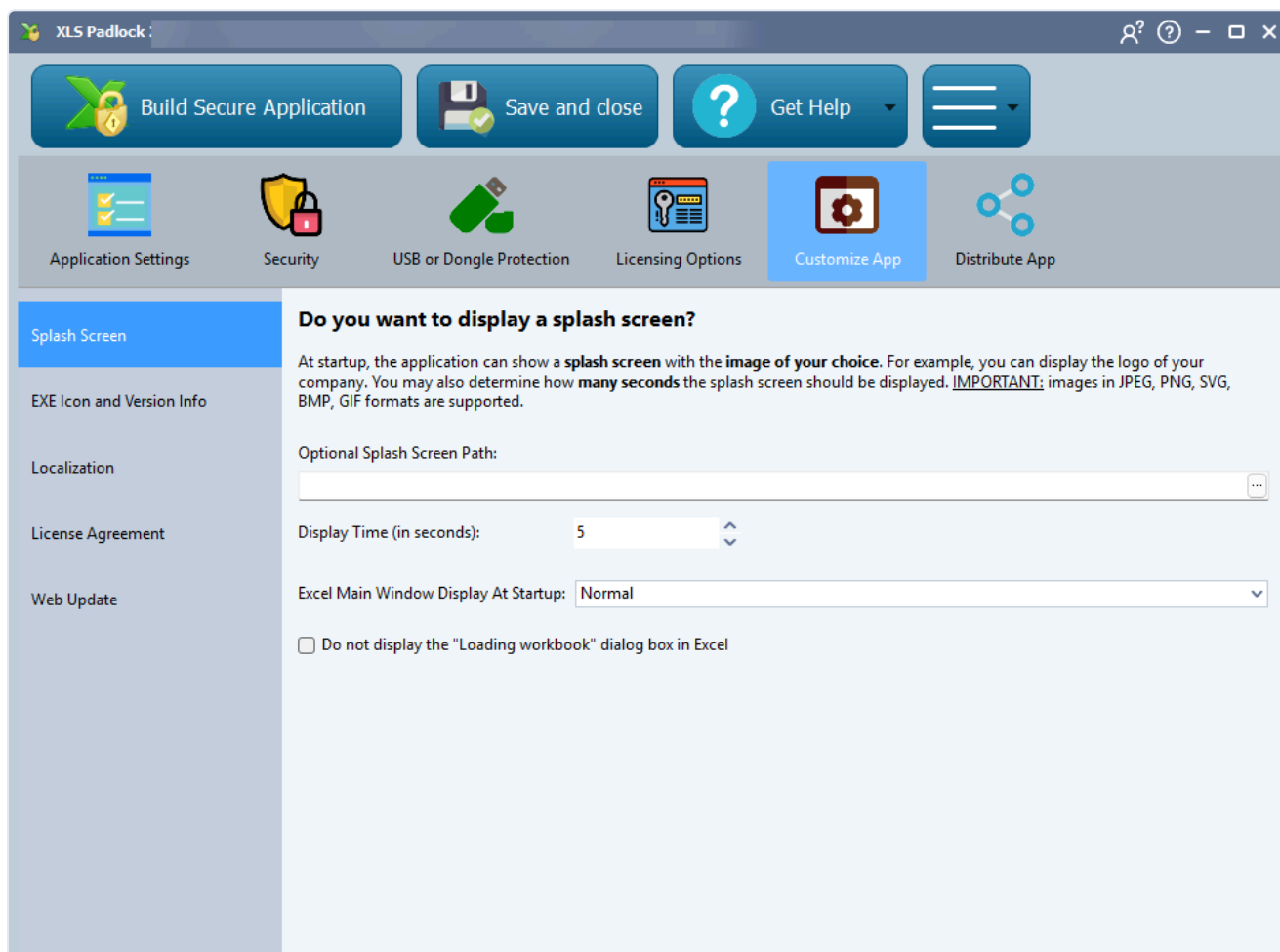
Le fichier d'icône doit être disponible en tant que fichier externe sur votre ordinateur pendant le processus de compilation.

**Info**

XLS Padlock appliquera également cette icône au fichier EXE de l'application autonome, garantissant une identité de marque cohérente.

# Fenêtre Excel au démarrage

L'option **Excel Main Window Display at Startup** (affichage de la fenêtre principale d'Excel au démarrage) vous permet de définir la taille de la fenêtre Excel au démarrage de votre application. Ce paramètre se trouve sur la page [Splash Screen](#).



Vous pouvez choisir parmi trois options :

- **Normal** : la fenêtre Excel s'ouvre avec sa taille et sa position par défaut. C'est le comportement par défaut.
- **Minimized** : l'application démarre réduite dans la barre des tâches de Windows. C'est utile si votre application s'exécute en arrière-plan ou si vous souhaitez afficher un formulaire personnalisé avant de montrer la fenêtre principale d'Excel.
- **Maximized** : la fenêtre Excel s'ouvre agrandie pour remplir tout l'écran.

### Besoin de masquer complètement la fenêtre Excel ?

Si votre application est pilotée uniquement par du code VBA et des UserForms et que vous voulez que la fenêtre Excel, ainsi que son icône dans la barre des tâches, n'apparaisse jamais du tout, consultez [Exécuter en tant qu'application VBA uniquement \(fenêtre principale d'Excel entièrement masquée\)](#). Lorsque cette option est activée, le choix ci-dessus est ignoré et Excel démarre toujours masqué.

## Voir aussi

---

- [Comment configurer le Splash Screen](#)
- [Comment masquer la boîte de dialogue « Loading workbook »](#)
- [Exécuter en tant qu'application VBA uniquement \(fenêtre principale d'Excel entièrement masquée\)](#)

# Exécuter en application VBA uniquement

---

Si votre Classeur protégé est entièrement piloté par du **code VBA** et des **UserForms**, et que la feuille de calcul elle-même n'est jamais destinée à être vue par vos utilisateurs finaux, vous pouvez le distribuer comme une *application VBA uniquement*. La fenêtre principale d'Excel reste masquée du début à la fin, y compris son icône dans la barre des tâches, et seul votre UserForm apparaît à l'écran. Votre application a l'apparence et le comportement d'un programme Windows autonome, sans aucune trace visible d'Excel.

Cette option est disponible depuis **XLS Padlock 2026.0**.

## Quand l'utiliser

---

Utilisez le mode VBA uniquement lorsque **toutes** les conditions suivantes sont réunies :

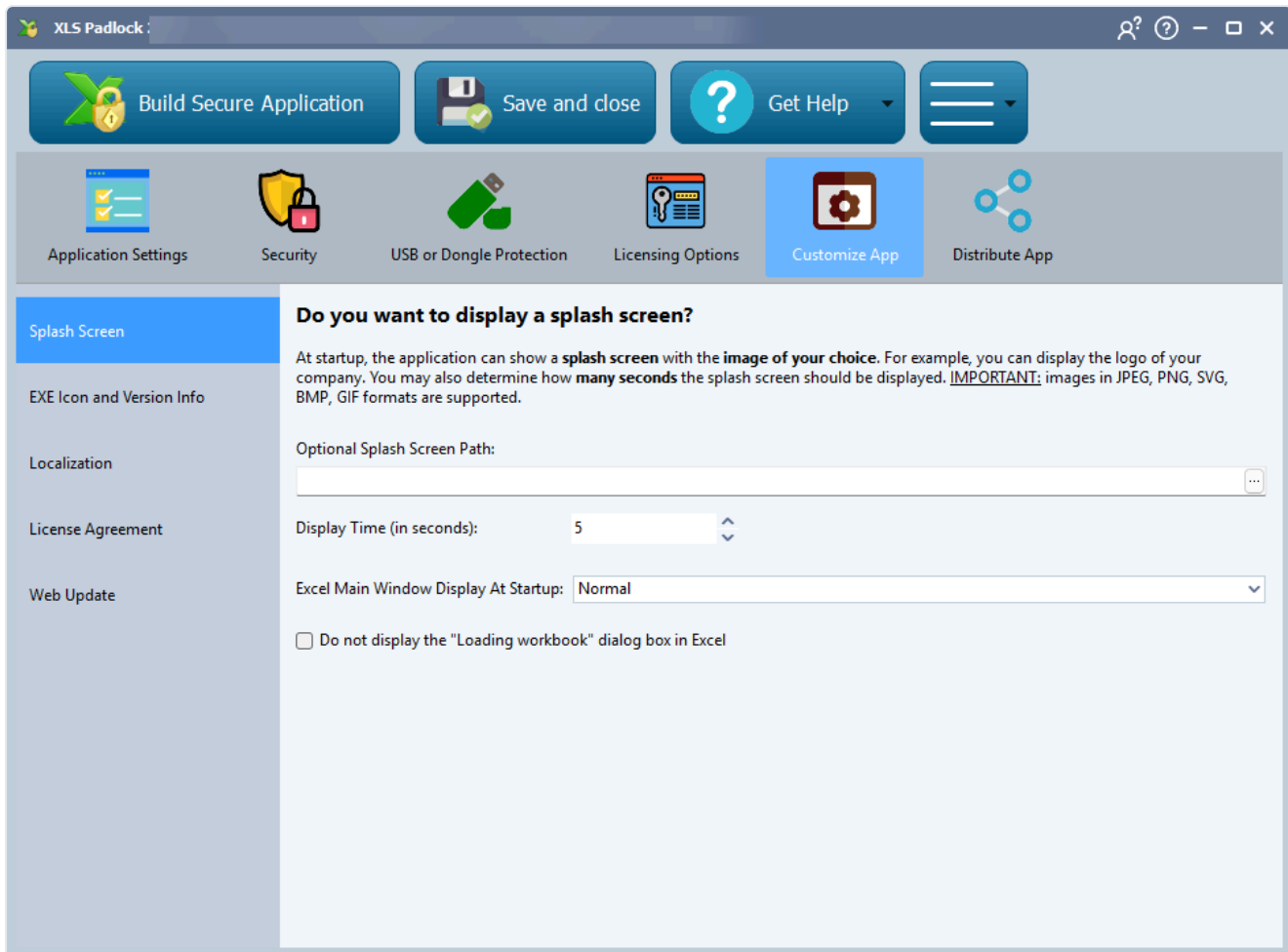
- L'interface utilisateur de votre application est construite avec des **UserForms**, et non avec des feuilles de calcul.
- Vous ne voulez pas que les utilisateurs finaux voient, modifient ou interagissent avec la feuille de calcul sous-jacente.
- Vous voulez une présentation soignée et personnalisée au démarrage, sans écran de démarrage Excel, sans entrée Excel dans la barre des tâches.

Si votre application repose sur des feuilles de calcul visibles (grilles de saisie de données, tableaux de bord rendus sous forme de cellules, etc.), n'activez **pas** cette option, les feuilles de calcul ne seront pas visibles pour vos utilisateurs.

## Comment l'activer

---

1. Ouvrez votre projet dans XLS Padlock.
2. Accédez à la page **Splash Screen** (sous Application Customization).
3. Cochez l'option **"Run as a VBA-only app (Excel main window fully hidden)"**.



C'est tout du côté de XLS Padlock. Vous devez maintenant ajouter un petit morceau de code VBA à votre Classeur afin que votre UserForm s'affiche au démarrage de l'application.

## Ajouter le code VBA de démarrage

Dans l'éditeur VBA d'Excel, ouvrez le module `ThisWorkbook` et collez :

```
Private Sub Workbook_Open()
 Application.Visible = False
 UserForm1.Show
End Sub
```

Remplacez `UserForm1` par le nom du formulaire que vous souhaitez afficher en premier.

### Pourquoi les deux, la case à cocher \*et\* le code VBA ?

La case à cocher de XLS Padlock garantit que la fenêtre principale d'Excel ne devient jamais visible **avant** l'exécution de votre code VBA (pas de flash, pas d'icône dans la barre des tâches). La ligne `Application.Visible = False` dans `Workbook_Open` maintient Excel masqué **après** la fin de l'initialisation d'Excel, Excel réaffirme sa visibilité en interne pendant le démarrage, le VBA doit donc imposer la règle une fois le Classeur entièrement chargé. L'utilisation des deux couches vous offre une expérience utilisateur propre et sans scintillement.

## Ce que voit l'utilisateur final

Avec la case à cocher activée et l'extrait VBA ci-dessus :

- **Aucun écran de démarrage Excel** au lancement.
- **Aucune fenêtre principale d'Excel** à aucun moment de la durée de vie de l'application.
- **Aucune entrée « Excel » dans la barre des tâches de Windows**, seulement votre UserForm.
- Votre application se comporte comme un programme Windows autonome construit autour de votre formulaire.

Si vous avez également configuré un [splash screen](#) et masqué la [boîte de dialogue Loading workbook](#), toute la séquence de démarrage reste dans votre interface personnalisée dès la toute première image.

## Fermeture de l'application

Comme Excel est masqué, le bouton de fermeture Excel standard n'est pas disponible pour vos utilisateurs. Vous fermez généralement l'application depuis votre UserForm à l'aide de :

```
Private Sub btnClose_Click()
 Application.Quit
End Sub
```

Ou si vous souhaitez laisser Excel s'exécuter tout en déchargeant votre interface :

```
Unload Me
```

## Interaction avec d'autres paramètres

Paramètre	Comportement lorsque « VBA-only app » est activé
<b>Excel Main Window Display at Startup</b> (Normal / Maximized / Minimized)	Ignoré, Excel démarre toujours masqué.
<b>Do not display the "Loading workbook" dialog box</b>	Recommandé de l'activer également, pour un démarrage entièrement silencieux.

Paramètre	Comportement lorsque « VBA-only app » est activé
<b>Splash Screen</b>	Fonctionne tel que configuré. Affiché avant même qu'Excel ne charge.

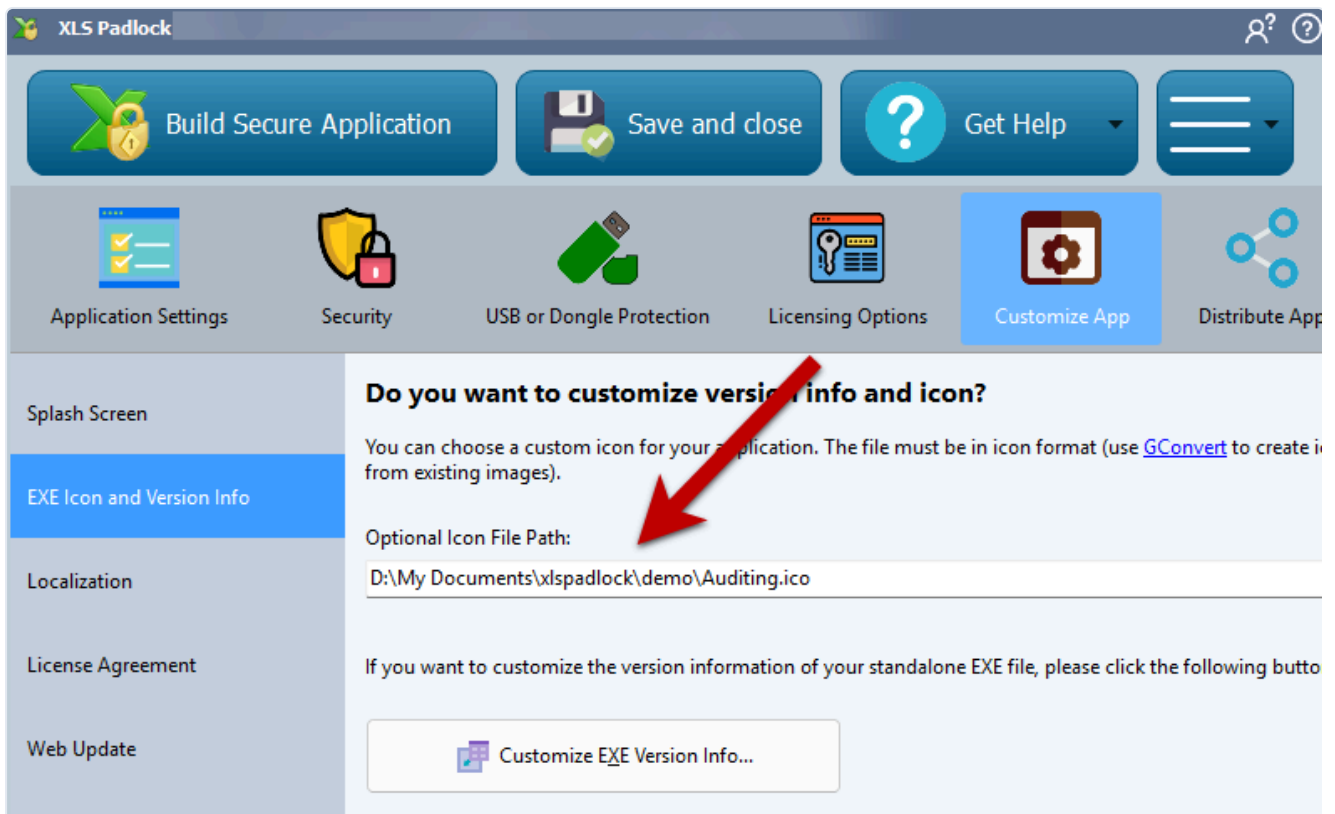
## Voir aussi

---

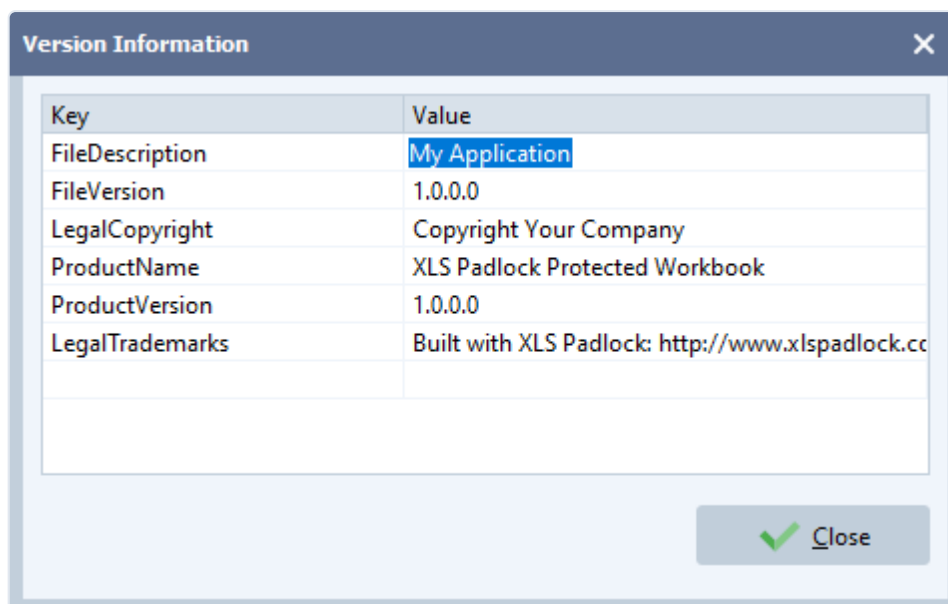
- [Comment configurer le Splash Screen](#)
- [Comment définir l'affichage de la fenêtre principale d'Excel au démarrage](#)
- [Comment masquer la boîte de dialogue « Loading workbook »](#)
- [VBA API Cookbook](#)

# Informations de version de l'EXE

Les **informations de version** d'un fichier exécutable forment une section de ressources spéciale contenant des détails comme le numéro de version du fichier, le système d'exploitation cible, le nom de fichier d'origine et les informations de copyright. Ces données sont intégrées dans l'EXE compilé. Lorsqu'elles sont incluses, les utilisateurs peuvent les consulter en faisant un clic droit sur l'icône du programme, en sélectionnant « Propriétés », puis en accédant à l'onglet « Détails » (ou en appuyant sur **Alt+Enter** sur le fichier dans l'Explorateur).



XLS Padlock vous permet d'intégrer vos propres informations de version dans le fichier EXE autonome. Cliquez sur « **Customize EXE Version Info** » pour ouvrir une fenêtre comportant les champs suivants :



- **File Description** : une brève description du contenu de votre application.
- **Company Name** : le nom de votre entreprise.
- **File Version** : le numéro de version de votre fichier .exe , au format X.X.X.X (par exemple, 1.20.34.45 ). Cette valeur est également utilisée par la [fonction de mise à jour Web](#) pour vérifier la présence de nouvelles versions de l'application.
- **Legal Copyright** : votre mention de copyright, par exemple « Copyright © [année] Your Company. All rights reserved. »
- **Product Name** : le nom de votre produit ou application, qui est généralement identique au titre de votre application.
- **Product Version** : le numéro de version de votre produit. Il correspond souvent à la File Version.
- **Legal Trademarks** : toutes les marques déposées que vous souhaitez inclure.

#### Info

La File Version est utilisée par la [fonction de mise à jour Web](#).

# Commutateurs de ligne de commande

Le fichier EXE de votre application compilée prend en charge plusieurs **commutateurs de ligne de commande** (également appelés arguments) qui automatisent certaines actions au démarrage. Vous pouvez les utiliser dans des raccourcis ou des scripts afin de contrôler le comportement de l'application à son lancement.

## CONSEIL

👉 Ces arguments de ligne de commande peuvent également être récupérés à l'exécution à l'aide de l'API VBA pour lire les paramètres de ligne de commande.

## Commutateurs disponibles

### -deact

Lance le processus de désactivation, permettant à un utilisateur de déplacer sa licence vers un autre ordinateur.

```
MYAPP.EXE -deact
```

### -deL

Efface tous les fichiers de sauvegarde sécurisés de l'application et charge le classeur d'origine. **À utiliser avec précaution, car cette action est irréversible.**

```
MYAPP.EXE -deL
```

### -enterkey

Ouvre la boîte de dialogue d'activation, permettant à l'utilisateur de saisir une nouvelle clé d'activation. Cela est utile pour remplacer une clé ancienne ou expirée.

```
MYAPP.EXE -enterkey
```

### -load

Ouvre une boîte de dialogue de fichier invitant l'utilisateur à sélectionner un fichier de sauvegarde à charger.

```
MYAPP.EXE -load
```

**-reset**

Abandonne toutes les modifications non enregistrées de la dernière session et charge le classeur d'origine non modifié. Cela ne supprime aucun fichier de sauvegarde `.xlsc`.

```
MYAPP.EXE -reset
```

**-webupdate**

Force l'application à vérifier immédiatement en ligne la présence d'une nouvelle version, selon vos paramètres « Web Update ».

```
MYAPP.EXE -webupdate
```

## Chargement d'un fichier de sauvegarde spécifique

---

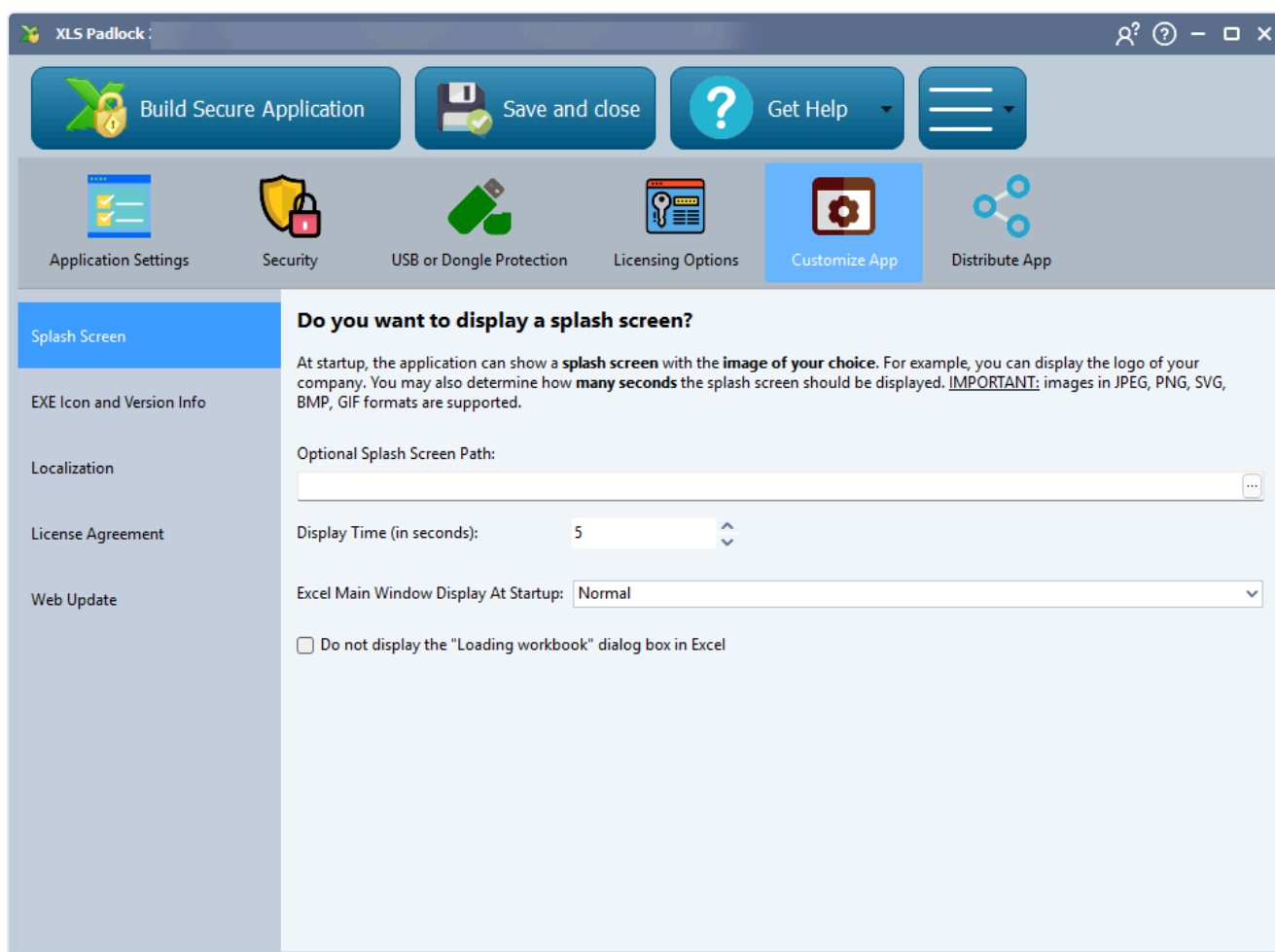
Vous pouvez également lancer l'application avec le chemin complet d'un fichier de sauvegarde sécurisé ( `.xlsc` ou `.xlsce` ) en argument. Cela chargera automatiquement le fichier spécifié au démarrage, sans aucune invite.

```
MyApp.exe "D:\My Documents\123.xlsc"
```

# Masquer la boîte de dialogue de chargement

Par défaut, lorsque votre application protégée démarre, Excel affiche une petite boîte de dialogue avec le message « **Loading workbook, please wait...** ».

Vous pouvez masquer cette boîte de dialogue en activant l'option « **Do not display the "Loading workbook" dialog box in Excel** », située sur la page [Splash Screen](#). Cela est utile pour offrir une expérience de démarrage plus épurée, en particulier lorsque vous utilisez déjà un écran de démarrage personnalisé.



Il est également possible de [masquer ce message en amont à l'aide de code VBA](#), ce qui vous donne davantage de contrôle sur le moment où la boîte de dialogue disparaît.

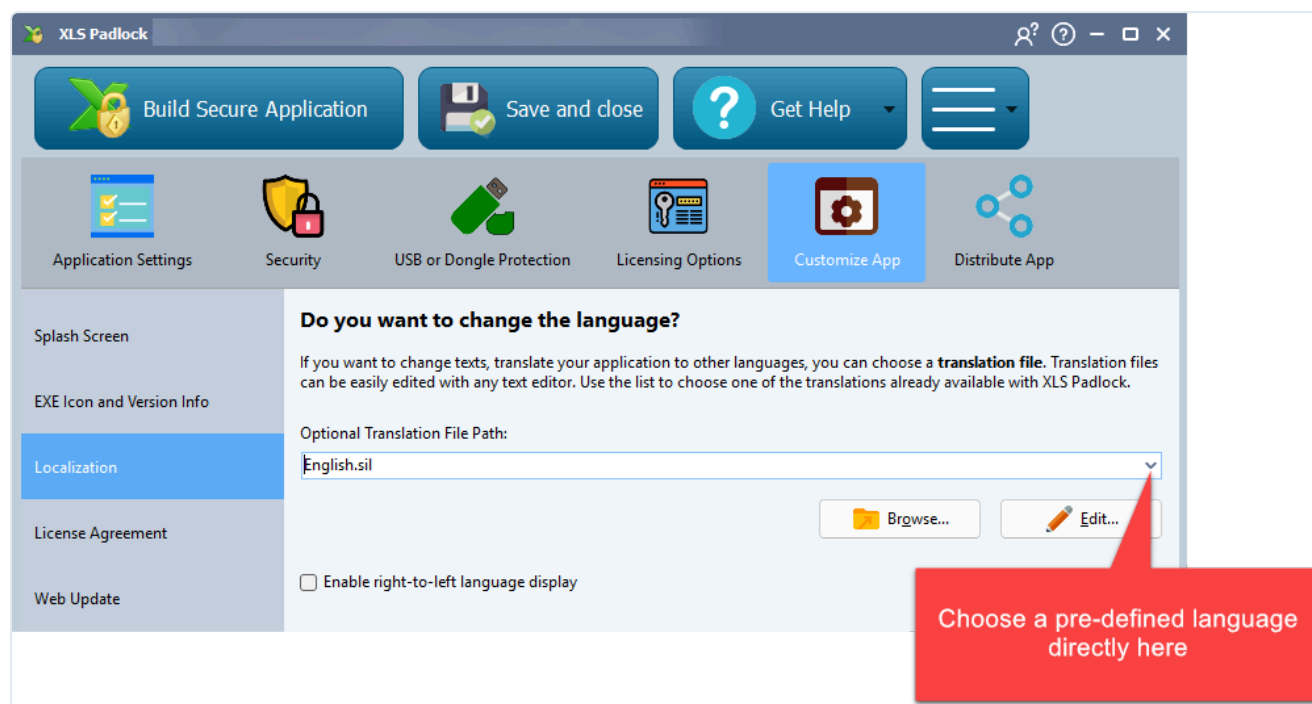
👉 Voir aussi

- [Comment configurer le Splash Screen](#)
- [Comment définir l'affichage de la fenêtre principale d'Excel au démarrage](#)

# Localisation et traduction

XLS Padlock offre une **prise en charge de la localisation** pour vos classeurs protégés, vous permettant d'afficher tout le texte de l'interface (titres de boîtes de dialogue, messages, invites et avertissements d'exécution) dans la langue de votre choix.

Vous pouvez configurer la localisation de votre application de classeur protégé ici :



## Format du fichier de traduction

Depuis XLS Padlock 2026, les fichiers de traduction utilisent le format **GNU gettext standard** avec l'extension `.po` (et son équivalent binaire compilé `.mo`). Il s'agit du même format que celui utilisé par les distributions Linux, Python, PHP, WordPress et d'innombrables autres applications, ce qui signifie que vous pouvez modifier les traductions avec l'un des nombreux outils gettext matures disponibles, dont l'éditeur libre et multiplateforme [Poedit](#).

L'application compilée incorpore une seule ressource `.mo` au moment de la compilation, chargée entièrement depuis la mémoire à l'exécution : aucun fichier temporaire n'est extrait sur le disque.

XLS Padlock est livré avec des fichiers `.po` pré-traduits pour **10 langues**, disponibles dans le sous-dossier `locale\` du répertoire d'installation de XLS Padlock :

Fichier	Langue
<code>en.po</code>	Anglais
<code>fr.po</code>	Français
<code>es.po</code>	Espagnol
<code>pt.po</code>	Portugais (Brésil)

Fichier	Langue
nL.po	Néerlandais
de.po	Allemand
ar.po	Arabe (disposition de droite à gauche prise en charge)
it.po	Italien
zh_CN.po	Chinois simplifié
zh_TW.po	Chinois traditionnel

## Demander une nouvelle langue

Si vous avez besoin d'une langue qui ne figure pas dans la liste, veuillez [nous contacter](#). Vous pouvez également créer votre propre fichier `.po` à partir de l'un des fichiers ci-dessus comme point de départ : consultez *Modification des fichiers de traduction* ci-dessous.

## Comment définir une langue pour votre classeur protégé

- Sur la page **Localization**, cliquez sur **Browse** à côté du champ *Translation file*. La boîte de dialogue s'ouvre par défaut sur le dossier `local\` de votre installation XLS Padlock, où se trouvent les 10 fichiers `.po` prêts à l'emploi. Choisissez la langue souhaitée, ou accédez à un fichier `.po` personnalisé de votre choix.
- Vous pouvez également coller le chemin complet manuellement dans le champ *Translation file*, ou le laisser vide pour utiliser la traduction anglaise par défaut incorporée dans XLS Padlock.

### IMPORTANT

Le fichier de traduction sélectionné doit être accessible à son chemin spécifié **lorsque vous compilez votre application**. XLS Padlock lit et compile ce fichier durant le processus de compilation et incorpore la ressource `.mo` résultante directement dans l'EXE protégé : il n'y a aucune dépendance à l'exécution vis-à-vis du fichier `.po` d'origine.

## Modification des fichiers de traduction

Les fichiers `.po` sont de simples fichiers texte UTF-8. Vous pouvez les modifier de deux manières :

- Avec Poedit** ([poedit.net](https://poedit.net), gratuit, multiplateforme) : recommandé. Il affiche les chaînes sources et traduites côte à côte, signale les entrées approximatives (fuzzy) et non traduites, valide la syntaxe à l'enregistrement et produit une sortie UTF-8 propre.
- Avec n'importe quel éditeur de texte** (Notepad++, VS Code, etc.). Enregistrez au format **UTF-8 sans BOM** et conservez une paire `msgid / msgstr` par bloc. Les lignes commençant par `#` sont des commentaires et ne sont pas affichées à l'utilisateur final.

Nous vous recommandons de **copier l'un des fichiers `.po` pré-fabriqués** vers un nouvel emplacement (à côté de votre classeur par exemple), de modifier la copie, puis de diriger XLS Padlock vers votre

nouveau fichier. La modification des fichiers situés dans le dossier d'installation de XLS Padlock nécessite généralement des droits d'administrateur et peut être écrasée par une future mise à jour.

## Migration depuis les anciens fichiers de traduction `.sil`

Si vous avez personnalisé l'interface utilisateur de votre classeur protégé sous XLS Padlock 2025.3 ou antérieur, votre traduction était stockée sous forme de fichier `.sil` (l'ancien format de traduction textuel utilisé par les versions précédentes). XLS Padlock 2026 ne lit plus les fichiers `.sil` au moment de la compilation : le format a été remplacé par `.po`.

Pour éviter de tout retaper, la distribution de XLS Padlock inclut un petit **assistant de migration SIL** → **PO** sous `tools\sil_to_po\` dans l'arborescence des sources :

```
python tools/sil_to_po/sil2po_customer.py your_translation.sil --lang fr --stats
```

Ceci produit `customer_fr.po` dans le dossier courant, automatiquement fusionné avec le fichier `.po` canonique actuel de sorte que :

- la colonne correspondant à votre langue cible soit reprise,
- les chaînes ajoutées depuis la rédaction de votre `.sil` apparaissent avec un `msgstr` vide (prêtes à être remplies dans Poedit),
- les chaînes dont la source anglaise a légèrement dérivé d'une version à l'autre sont signalées `#, fuzzy` pour votre vérification,
- les chaînes qui n'existent plus dans XLS Padlock sont ajoutées à la fin sous forme d'entrées obsolètes `#~` (ou supprimées entièrement avec `--drop-orphans`).

Pour migrer plusieurs langues en une seule exécution, séparez-les par des virgules :

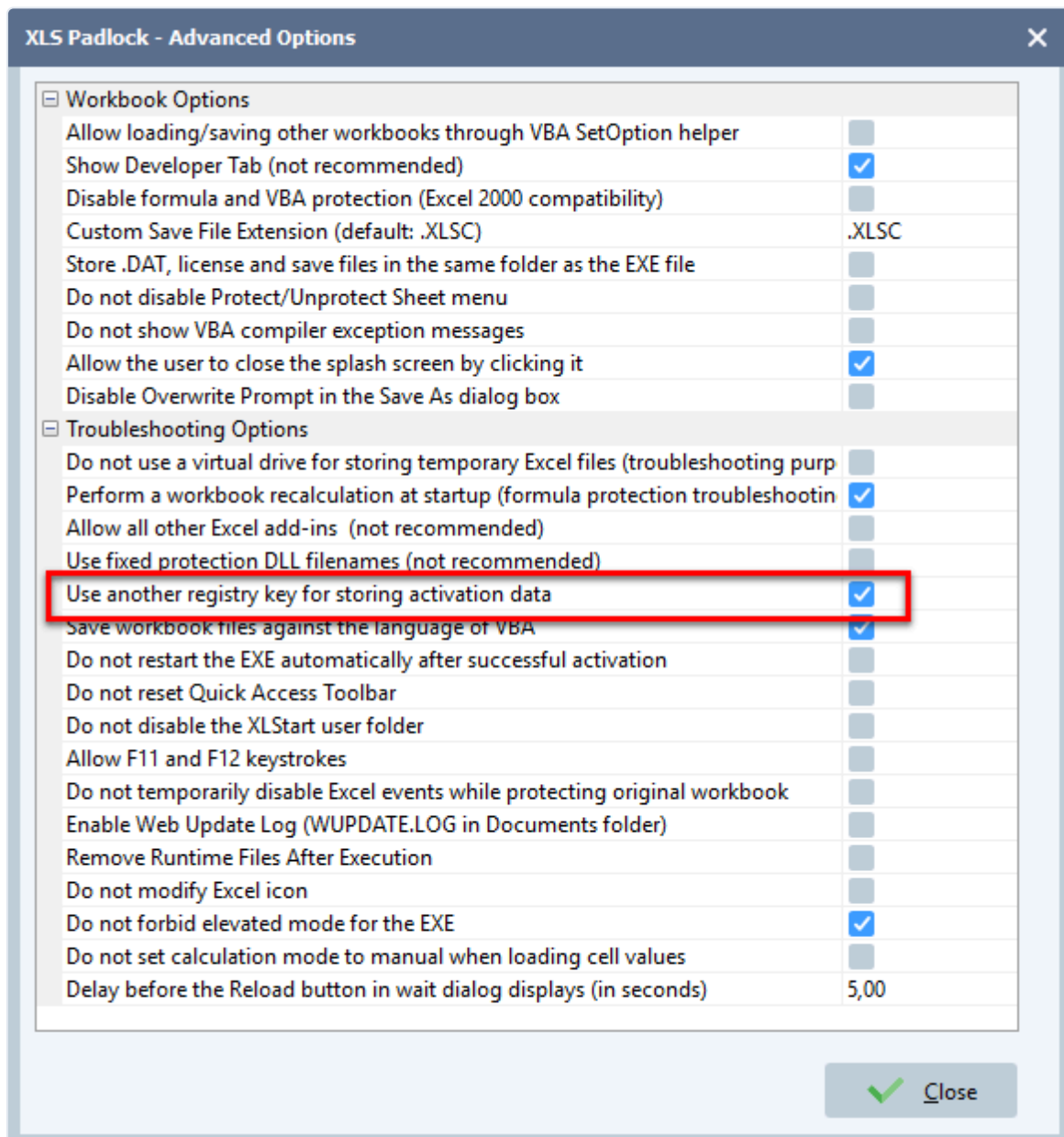
```
python tools/sil_to_po/sil2po_customer.py your_translation.sil --lang fr,es,de --stats
```

Les fichiers `.po` résultants sont immédiatement utilisables depuis le sélecteur *Translation file* dans le XLS Padlock Designer. Le script assistant et son fichier compagnon `README.md` (référence complète des options, algorithme de correspondance, exemples détaillés) sont livrés dans le sous-dossier `tools\sil_to_po\` du répertoire d'installation de XLS Padlock : ouvrez `README.md` dans n'importe quel éditeur de texte pour lire la documentation complète localement.

Python 3.8 ou ultérieur est requis pour exécuter l'assistant. Aucune dépendance Python externe : le script est autonome.

# Options avancées

Dans « Application Settings », cliquez sur le bouton « Configure Advanced Options » pour afficher la fenêtre suivante :



## Réservé aux utilisateurs avancés

Ces options sont destinées aux utilisateurs avancés. Ne les modifiez que si vous comprenez leur objectif ou si notre support technique vous le conseille.

### ## Options avancées disponibles

- **Allow loading/saving other workbooks through VBA SetOption helper** : un contournement de sécurité pour des opérations VBA spécifiques. Consultez [Chargement/enregistrement de classeurs via](#)

### [l'assistant VBA SetOption.](#)

- **Show Developer Tab** : rend visible l'onglet Développeur d'Excel (non recommandé pour la sécurité).
- **Disable formula and VBA protection** : pour la compatibilité avec Excel 2000.
- **Custom Save File Extension** : modifie l'extension `.XLSC` par défaut des fichiers de sauvegarde.
- **Store .DAT, license and save files in the same folder as the EXE file** : active le mode portable.
- **Do not disable Protect/Unprotect Sheet menu** : conserve activés les éléments de menu de protection de feuille.
- **Do not show VBA compiler exception messages** : masque les messages d'erreur détaillés du VBA Compiler. Consultez [Désactiver les informations de débogage](#).
- **Allow the user to close the splash screen by clicking it** : permet aux utilisateurs de passer l'écran de démarrage.
- **Disable Overwrite Prompt in the Save As dialog box** : empêche le message « do you want to overwrite? ».
- **Do not use a virtual drive for storing temporary Excel files** : à des fins de débogage.
- **Perform a workbook recalculation at startup** : pour le débogage de la protection des formules.
- **Allow all other Excel add-ins** : abaisse la sécurité en activant tous les compléments (non recommandé).
- **Use fixed protection DLL filenames** : non recommandé pour la sécurité.
- **Use another registry key for storing activation data** : une solution de contournement pour les conflits de registre.
- **Save workbook files against the language of VBA** : résout certains problèmes de localisation.
- **Do not restart the EXE automatically after successful activation** : empêche le redémarrage automatique.
- **Do not reset Quick Access Toolbar** : préserve les personnalisations de l'utilisateur apportées à la barre d'outils Accès rapide (QAT).
- **Do not disable the XLStart user folder** : autorise l'exécution des compléments présents dans le dossier XLStart de l'utilisateur.
- **Allow F11 and F12 keystrokes** : active ces touches de fonction.
- **Do not temporarily disable Excel events while protecting original workbook** : à des fins de débogage.
- **Enable Web Update Log** : crée un fichier `WUPDATE.LOG` dans le dossier Documents de l'utilisateur pour déboguer la fonctionnalité de mise à jour Web.
- **Remove Runtime Files After Execution** : nettoie les fichiers temporaires.
- **Do not modify Excel icon** : empêche XLS Padlock de modifier l'icône de la fenêtre Excel.
- **Do not forbid elevated mode for the EXE** : permet à l'application de s'exécuter en tant qu'administrateur.
- **Do not set calculation mode to manual when loading cell values** : à des fins de débogage.
- **Delay before the Reload button in wait dialog displays (in seconds)** : personnalise la boîte de dialogue d'attente.

# Ignorer les erreurs de traitement

---

L'option « Ignore errors when processing the workbook (internal protection) » permet au processus de compilation de se poursuivre même si des erreurs sont rencontrées lors de l'ouverture de votre classeur Excel. Cette fonctionnalité est conçue pour rationaliser le flux de protection en supprimant les messages d'erreur qui pourraient autrement interrompre le processus.

Lorsque cette option est activée, XLS Padlock ne s'arrête pas et n'affiche pas d'invites d'erreur s'il rencontre des problèmes durant la phase d'ouverture du classeur.

## **Cette option ne corrige pas les erreurs**

L'activation de cette option ne résout aucun problème sous-jacent du classeur ; elle ne fait que les contourner durant la compilation. Après avoir protégé votre classeur avec cette option activée, vous devez le tester minutieusement pour vous assurer que toutes les fonctionnalités fonctionnent comme prévu.

Si vous rencontrez toujours des erreurs dans le classeur compilé, envisagez d'activer l'option alternative, « [Utiliser l'automatisation Excel pour la protection des formules](#chapter-formula-protection-method) », qui peut atténuer les problèmes que la méthode de protection interne standard ne résout pas.

# Désactiver les informations de débogage

---

Par défaut, si le VBA compiler détecte une erreur dans le code compilé à l'exécution, il affiche un message d'erreur détaillé pour vous aider à déboguer le problème.

Si vous ne souhaitez pas afficher ces informations à vos utilisateurs finaux, vous pouvez les désactiver. Rendez-vous dans [Configure Advanced Options](#) et désactivez l'option : « **Do not show VBA compiler exception messages** ».

# Interface Excel personnalisée

---

Comme l'explique la documentation de Microsoft, vous pouvez personnaliser l'interface utilisateur d'Excel en plaçant un fichier `.officeUI` spécialement conçu dans un dossier utilisateur précis.

XLS Padlock vous permet de spécifier votre propre Excel.officeUI pour l'application sécurisée. Il remplacera automatiquement l'interface par défaut de l'utilisateur (si elle est disponible).

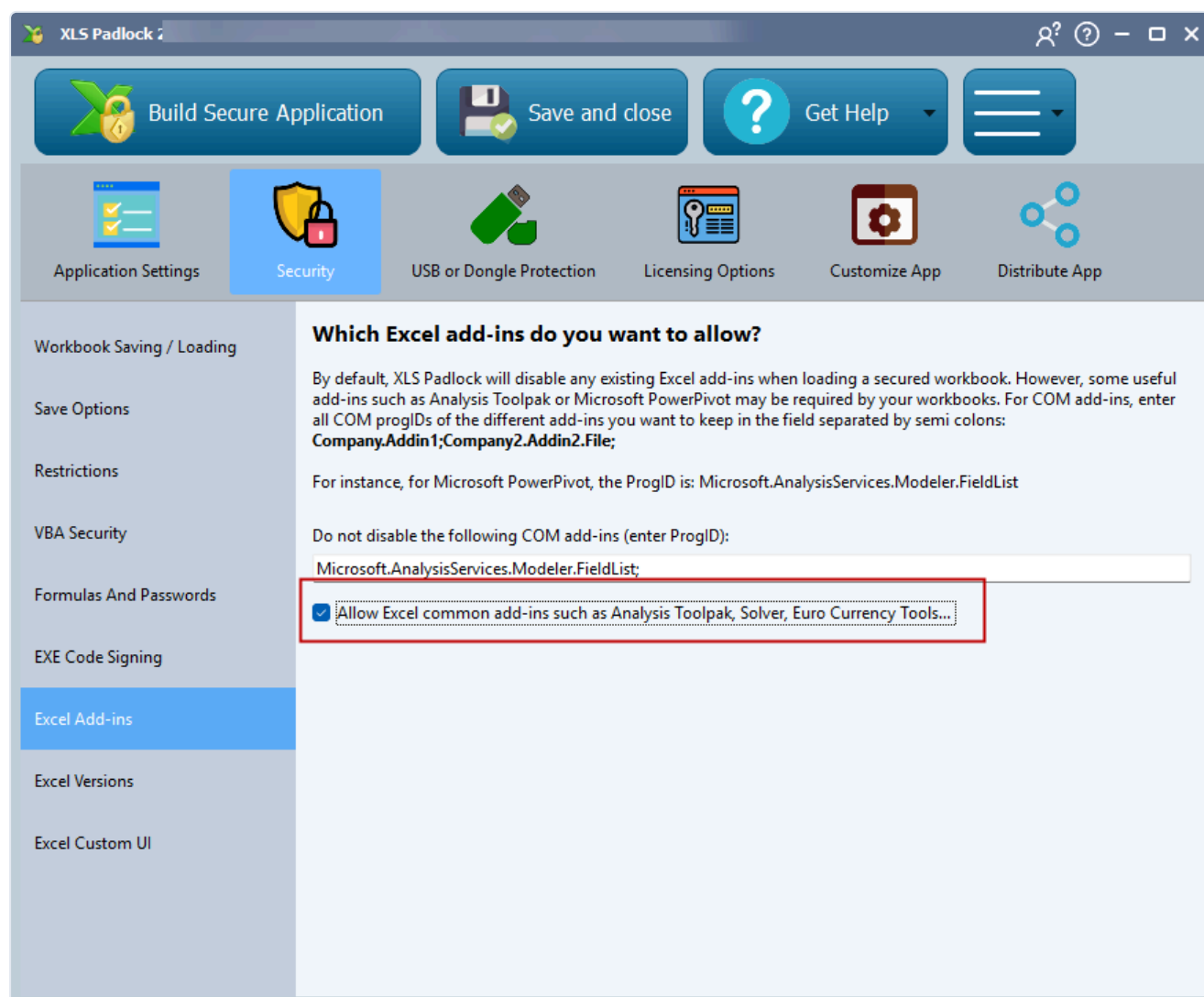
Il vous suffit de fournir le chemin complet vers votre fichier `.officeUI`. XLS Padlock le compilera directement dans l'EXE final, vous n'avez donc pas besoin de déployer le fichier `.officeUI` séparément.

Pour plus d'informations sur la création de fichiers XML d'interface personnalisée, veuillez consulter la [documentation officielle de Microsoft Office](#).

# Compléments Excel

Par défaut, XLS Padlock désactive la plupart des compléments Excel lorsqu'un classeur sécurisé est chargé. Il s'agit d'une mesure de sécurité destinée à créer un environnement contrôlé pour votre application.

Cependant, si votre classeur dépend de compléments spécifiques pour fonctionner correctement, vous pouvez les réactiver de manière sélective. Cette page explique comment autoriser à la fois les compléments COM et les compléments Excel intégrés courants.



## Conserver des compléments COM spécifiques activés

Si votre classeur requiert un complément COM (comme Microsoft PowerPivot), vous devez spécifier son ProgID pour empêcher XLS Padlock de le désactiver.

Dans le champ "Do not disable the following COM add-ins", saisissez le ProgID de chaque complément que vous souhaitez conserver activé. Si vous en avez plusieurs, séparez-les par des points-virgules ( ; ).

**Exemple :** Pour conserver Microsoft PowerPivot activé, vous saisissez ses ProgID :

```
Microsoft.AnalysisServices.Modeler.FieldList;PowerPivotExcelClientAddIn.NativeEntry.1
```

### Trouver un ProgID

Le ProgID (identifiant programmatique) est un nom unique stocké dans le registre Windows pour un composant COM. Vous pouvez souvent le trouver dans la documentation du complément ou en inspectant le registre. Microsoft fournit davantage d'informations sur ce sujet sur son site web d'assistance.

## Autoriser les compléments Excel courants

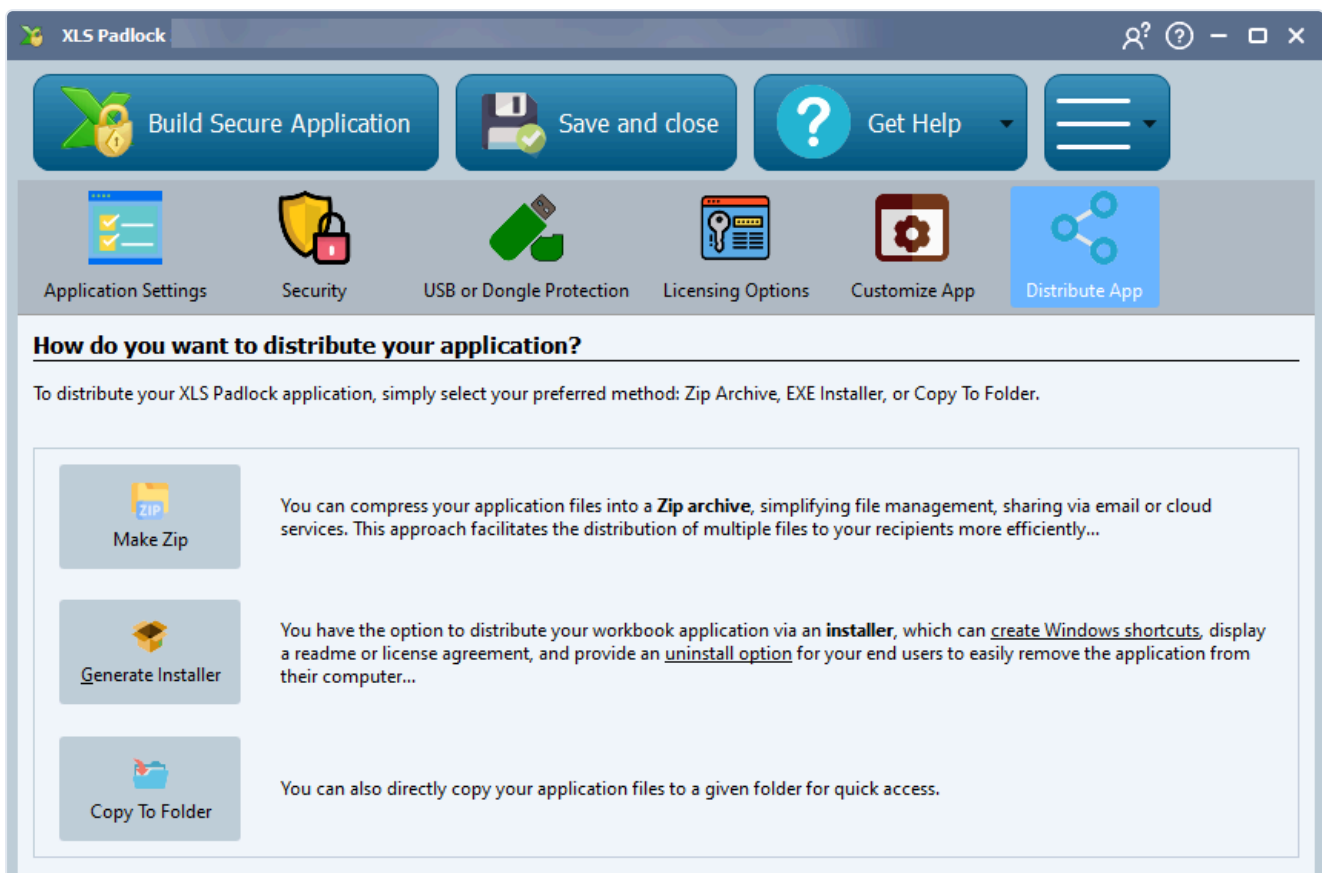
Excel inclut également plusieurs compléments intégrés populaires, tels que l'**Analysis ToolPak**, le **Solver** et les **Euro Currency Tools**.

Si votre classeur utilise des fonctionnalités de l'un de ces compléments, activez l'option "**Allow Excel common add-ins...**". Cela garantira leur chargement et leur disponibilité au démarrage de votre application protégée.

# Distribuer l'application

Une fois que vous avez [converti votre classeur en exécutable \(.EXE\)](#), l'étape suivante consiste à le distribuer à vos utilisateurs.

XLS Padlock propose plusieurs méthodes pour simplifier ce processus : créer une archive Zip, générer un programme d'installation EXE, ou copier directement les fichiers dans un dossier.



## Archive Zip

Une archive Zip est une méthode répandue et simple pour distribuer un logiciel. Compresser votre application dans un seul fichier `.zip` simplifie la gestion des fichiers pour vos utilisateurs. Cette méthode est particulièrement utile pour partager votre application par e-mail ou via des services cloud, car elle réduit la taille du fichier et regroupe tous les composants nécessaires dans un seul paquet.

Pour créer une archive Zip de votre application, cliquez sur **Make Zip** et choisissez l'emplacement où vous souhaitez enregistrer le fichier `.zip`.

## Programme d'installation EXE

Pour une touche plus professionnelle, envisagez de distribuer votre application avec un programme d'installation EXE. Cette méthode améliore l'expérience utilisateur en proposant un processus

d'installation familier, courant dans les environnements Windows. Un programme d'installation peut créer des raccourcis sur le Bureau et dans le menu Démarrer, afficher un fichier readme ou un contrat de licence, et fournir un désinstalleur, permettant aux utilisateurs de supprimer facilement l'application de leur ordinateur.

➔ Pour des instructions détaillées, consultez notre guide sur [comment créer un programme d'installation pour votre application](#).

## Copier vers un dossier

---

Enfin, copier directement les fichiers de votre application dans un dossier est la méthode de distribution la plus directe. Elle comporte deux étapes simples :

- Copier tous les fichiers nécessaires de l'application dans un dossier.
- Partager ce dossier avec vos utilisateurs via un support physique (comme une clé USB) ou par transfert réseau.

Bien que moins aboutie que les autres méthodes, cette approche est la manière la plus rapide de faire fonctionner votre application sur un autre système, car elle ne nécessite aucun utilitaire de compression ni logiciel d'installation particulier.

👉 Voir aussi :

- [Comment signer numériquement votre application](#)

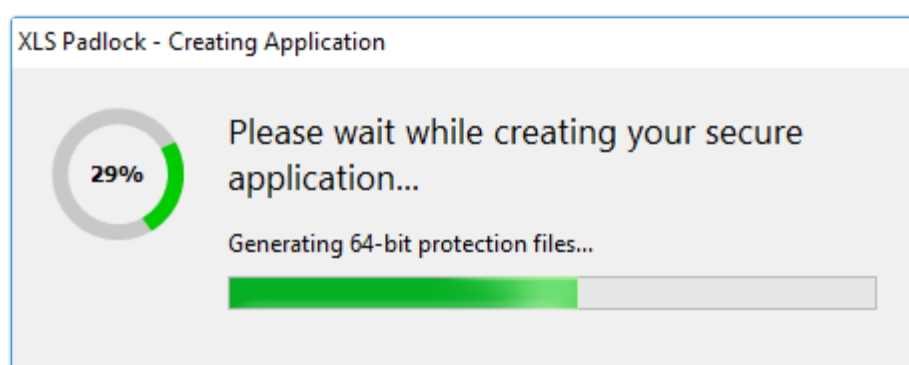
# Distribuer un classeur protégé

## Compiler votre classeur Excel en EXE

Dans XLS Padlock, lorsque vous cliquez sur « **Build Secure Application** »



, la boîte de dialogue de progression de la compilation s'affiche :



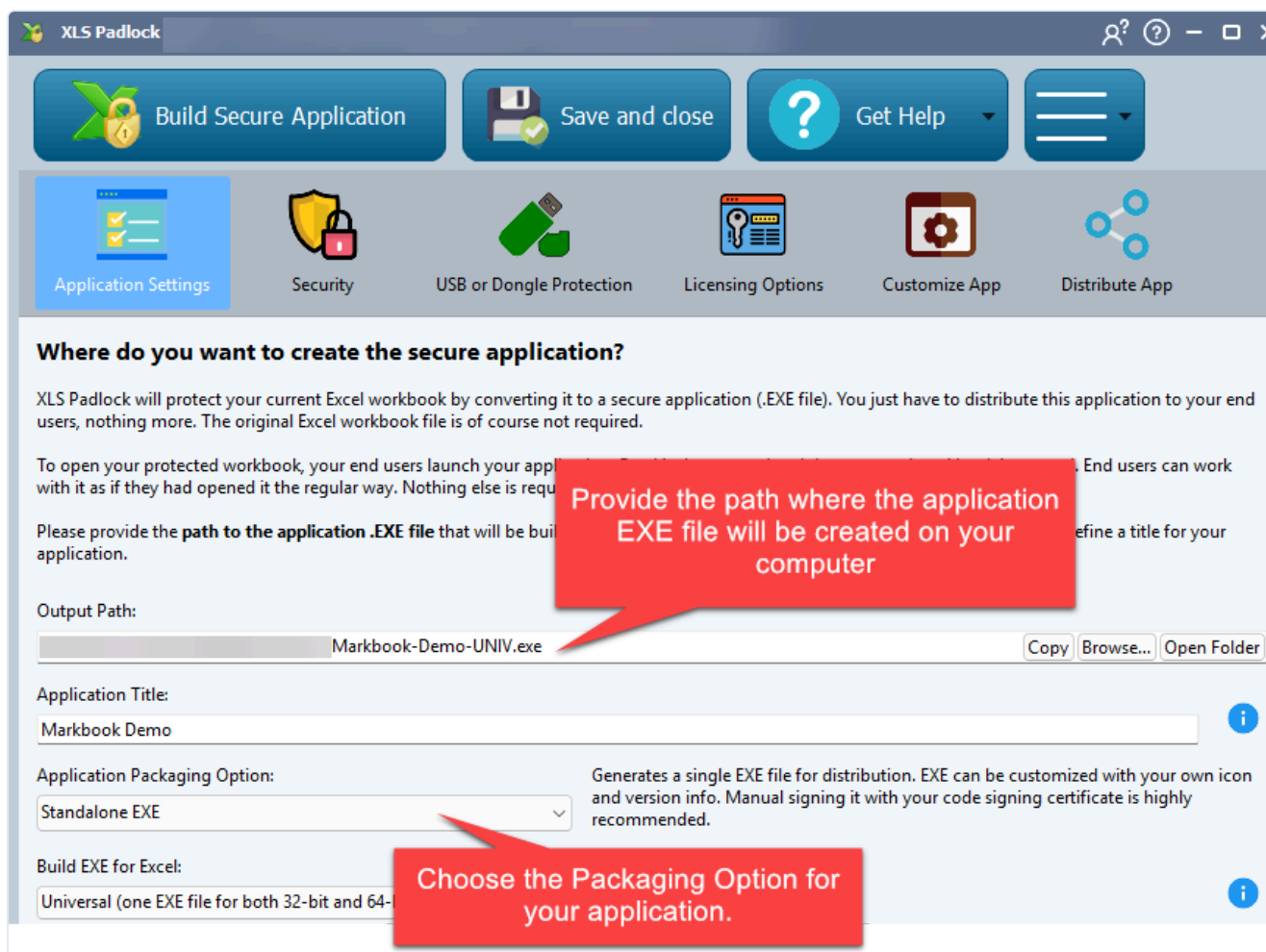
Une fois terminée, **votre application sécurisée est prête**. XLS Padlock a généré un **fichier exécutable** (.EXE) à partir de votre classeur Excel.

**► Désormais, pour accéder au classeur protégé, les utilisateurs finaux lancent ce fichier .EXE. Une copie locale de Microsoft Excel est la seule autre exigence.** Votre fichier de classeur d'origine n'est **PLUS** requis.

- Au lancement du fichier .EXE, Excel démarre et le classeur protégé s'ouvre. Les utilisateurs finaux peuvent interagir avec lui comme s'ils l'avaient ouvert de la manière habituelle.
- Certaines fonctionnalités d'Excel comme « New Workbook », Open et Save (en option) sont désactivées pour des raisons de sécurité.
- Si Excel n'est pas trouvé sur l'ordinateur de l'utilisateur, un message d'erreur s'affiche. Vous pouvez configurer votre application pour [exiger des versions spécifiques d'Excel](#).

## Options de packaging de l'application

XLS Padlock propose deux formats pour votre application sécurisée, ce qui influence la manière dont vous la distribuez et gérez les avertissements de sécurité comme Windows SmartScreen.



## EXE autonome

XLS Padlock crée un fichier exécutable unique (.EXE) à partir de votre classeur. Il vous suffit de distribuer ce fichier unique à vos utilisateurs finaux. Le classeur Excel d'origine n'est pas requis.

### La signature de code est fortement recommandée

Nous conseillons vivement de [signer numériquement vos fichiers EXE](#). Distribuer un fichier EXE non signé provoquera probablement l'affichage par Windows SmartScreen d'un avertissement « Unknown Application ». Bien que votre EXE fonctionne tout de même, une signature numérique apporte de la confiance et une meilleure expérience utilisateur.

### ### Lot d'application EXE + XPLAPP

Cette option génère un fichier EXE, un fichier compagnon `.bin64` et un fichier de données `.xplapp` distinct. L'EXE principal est pré-signé par notre société, ce qui facilite sa reconnaissance par Windows SmartScreen et les logiciels antivirus.

- Vous devez distribuer les trois fichiers ensemble ( `.exe` , `.bin64` et `.xplapp` ).
- XLS Padlock peut regrouper ces fichiers dans une [archive Zip unique](#) ou un [programme d'installation](#) pour faciliter la distribution.

- Cette option réduit considérablement le risque d'avertissements « Unknown Application » et de faux positifs antivirus.

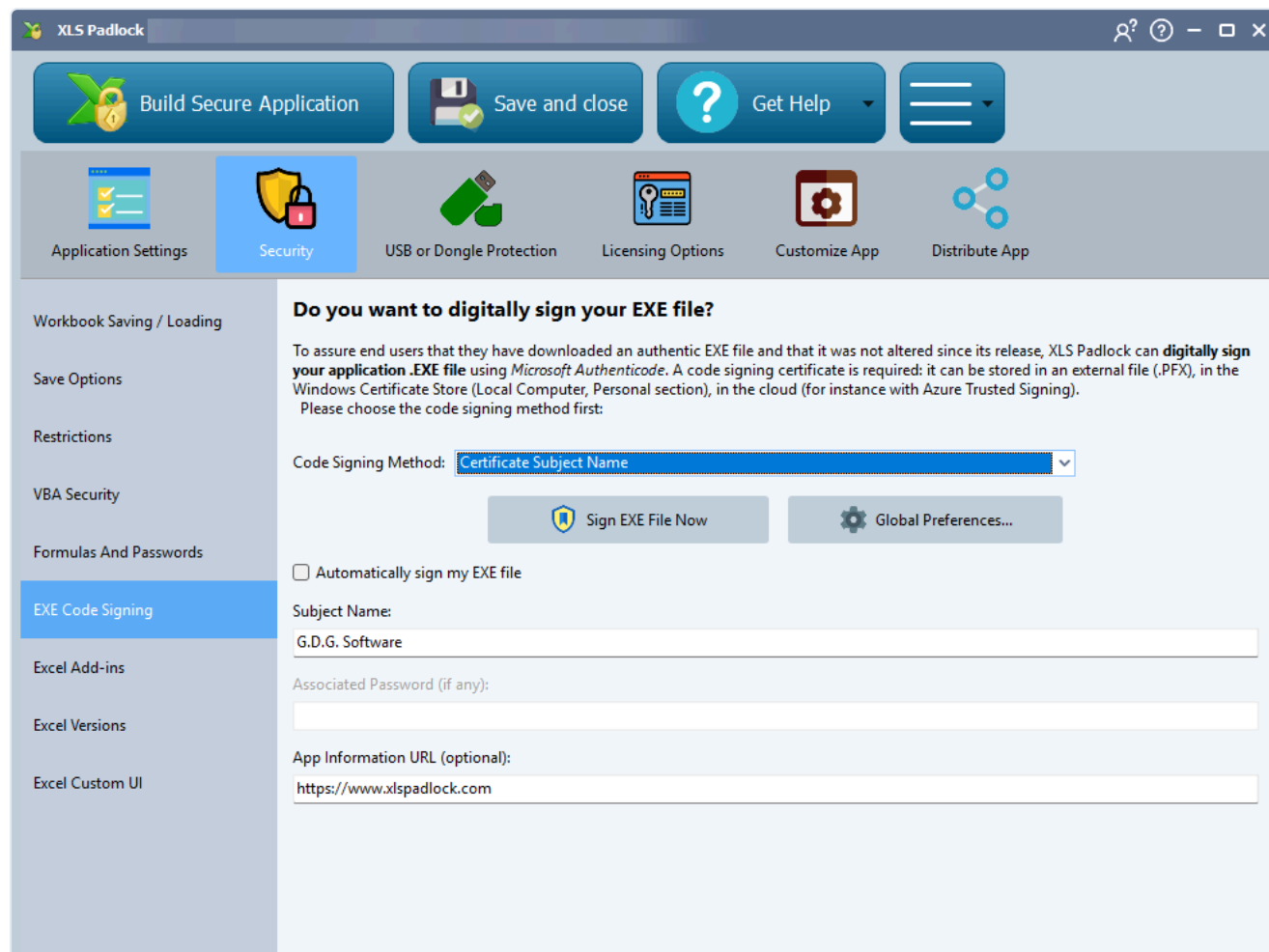
## Recommandations de distribution

---

1. **Signez numériquement vos fichiers EXE** : c'est l'étape la plus importante pour éviter les faux positifs antivirus et instaurer la confiance auprès de vos utilisateurs. XLS Padlock peut [automatiser ce processus](#) si vous disposez d'un certificat de signature de code.
2. **Désactivez temporairement votre antivirus** : si vous avez un programme antivirus actif, envisagez de le désactiver avant de compiler vos classeurs. Les logiciels antivirus peuvent parfois signaler à tort la création de nouveaux fichiers EXE sur votre ordinateur.
3. **N'envoyez pas les fichiers EXE par e-mail** : la plupart des fournisseurs de messagerie bloquent les fichiers `.exe`. À la place, téléversez les fichiers de votre application vers un service d'hébergement de fichiers comme Dropbox, Google Drive ou votre propre serveur web, et partagez le lien de téléchargement. Vous pouvez également [empaqueter votre application dans une archive Zip](#).

# Signer numériquement l'EXE

Lorsque vous **signez numériquement votre fichier EXE de classeur autonome**, vous assurez aux utilisateurs finaux que le fichier est authentique et n'a pas été altéré. Ce processus, également appelé signature de code, utilise la technologie Microsoft Authenticode® pour vérifier que le code provient d'un éditeur de confiance.



XLS Padlock simplifie le processus de signature de code en prenant en charge les étapes nécessaires en interne.

## Pourquoi la signature de code est importante

Si vous prévoyez de distribuer vos applications sur Internet, la signature de code est fortement recommandée. Elle permet d'éviter que les navigateurs web et Windows n'affichent des avertissements « Unidentified Publisher » et peut aussi réduire la probabilité de faux positifs émis par les logiciels antivirus.



## Comment obtenir un certificat de signature de code

Pour signer votre application, vous avez besoin d'un **Code Signing Certificate** valide délivré par une autorité de certification (CA) de confiance comme Sectigo ou Digicert. Les autres types de certificats, comme SSL/TLS, ne sont pas compatibles.

### Certificats basés sur le cloud

Une alternative de plus en plus populaire et économique est **Azure Trusted Signing**, un service basé sur le cloud de Microsoft. Il élimine le besoin de jetons USB physiques et simplifie le processus de gestion. XLS Padlock prend entièrement en charge la [signature avec Azure Trusted Signing](#).

### ## Certificats basés sur un jeton (HSM)

Depuis le 1er juin 2023, toutes les nouvelles clés privées de certificat de signature de code doivent être stockées sur du matériel sécurisé, comme un jeton USB conforme à la norme FIPS 140-2 niveau 2 ou un module de sécurité matériel (HSM). Cela renforce la sécurité en empêchant le vol de clés. XLS Padlock fonctionne parfaitement avec les certificats basés sur un jeton ; assurez-vous simplement que le jeton est branché sur votre ordinateur lorsque vous générez votre application.

## Configurer la signature de code dans XLS Padlock

Dans l'interface de XLS Padlock, accédez à l'onglet **Security -> EXE Code Signing**. Pour activer la signature, choisissez votre **Code Signing Method** préférée :

- **PFX File** : utilise un certificat stocké dans un fichier `.pfx`. Il s'agit d'une méthode héritée pour les anciens certificats.
- **Certificate Subject Name** : localise le certificat dans le magasin de certificats Windows par son nom de sujet (Subject Name). C'est une méthode courante pour les certificats sur jetons matériels.
- **Certificate Thumbprint** : localise le certificat dans le magasin de certificats Windows par son empreinte unique (Thumbprint, un hachage SHA-1). C'est souvent la méthode la plus fiable.
- **SignTool Commands** : une méthode avancée qui vous permet de fournir des commandes personnalisées pour l'utilitaire `SignTool.exe` de Microsoft, offrant un maximum de flexibilité.
- **Azure Trusted Signing** : signe votre application avec le service basé sur le cloud de Microsoft. Consultez notre [tutoriel Azure Trusted Signing](#) pour plus de détails.

Azure CLI requise

Pour utiliser cette méthode, vous devez d'abord installer la Microsoft Azure CLI et vous connecter à l'aide de `az login`.

## Effectuer la signature

- **Signature manuelle** : cliquez sur **Sign EXE File Now** pour signer immédiatement le dernier fichier EXE généré.
- **Signature automatique** : cochez **Automatically sign my EXE file** pour que XLS Padlock signe l'EXE à chaque fois que vous générez votre application.

### Dépannage

Si une erreur de signature se produit, vérifiez les messages détaillés dans le journal de compilation de XLS Padlock. Ce fichier journal est généralement nommé [Your Workbook Filename].xplcompil.log et se trouve dans le même répertoire que votre classeur.

# Créer un programme d'installation

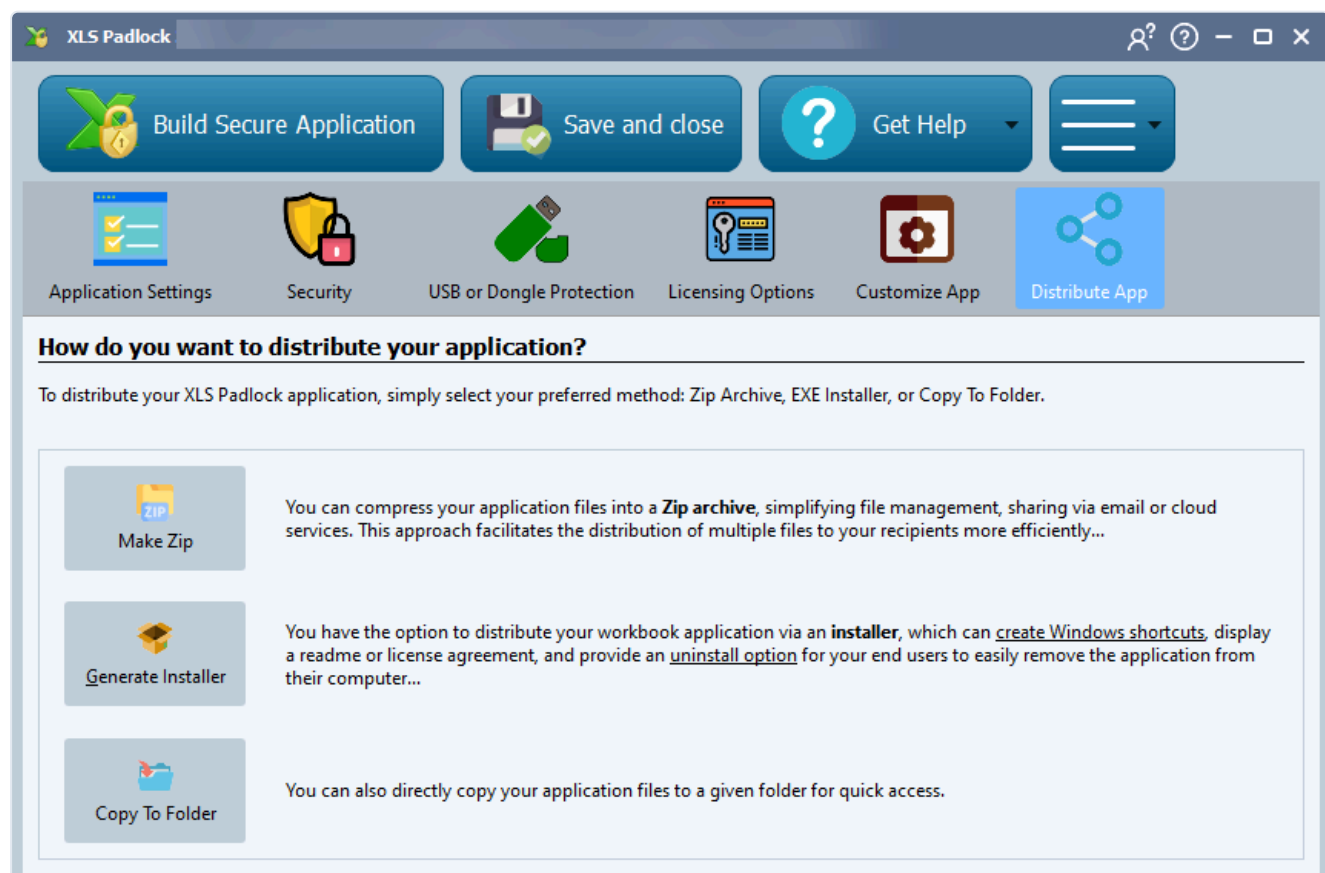
Empaqueter votre classeur compilé dans un programme d'installation professionnel (ou programme de configuration) est un excellent moyen de distribuer votre application. Les programmes d'installation offrent une expérience utilisateur familière et peuvent gérer des tâches telles que l'installation de plusieurs fichiers (comme un fichier readme), l'affichage d'un contrat de licence, la création de raccourcis et la fourniture d'un programme de désinstallation pour une suppression propre.

XLS Padlock s'intègre avec notre logiciel, [Paquet Builder](#), pour générer des programmes d'installation personnalisés et compacts. **Paquet Builder doit être installé sur votre ordinateur pour utiliser cette fonctionnalité.**

*Paquet Builder est un outil puissant qui combine un créateur d'archives auto-extractibles 7-Zip avec un générateur de routine de configuration. Il vous permet de créer des programmes d'installation flexibles et compacts pour une distribution logicielle professionnelle, d'empaqueter des documents et des fichiers de programme, de construire des packages d'installation multilingues, de générer des mises à jour et d'envelopper plusieurs fichiers dans un seul `.exe` pour une distribution facile en ligne.*

Pour créer un programme d'installation :

1. Cliquez sur **Generate Installer** dans l'onglet "Distribute" de XLS Padlock. Cela ouvrira la fenêtre "Make Setup for your compiled workbook".



1. Renseignez le "Destination path" (dossier d'installation par défaut), le "Setup title" (titre de la fenêtre du programme d'installation) et le "Your Application Name" (pour les raccourcis, le programme de

désinstallation, etc.).

- Appuyez sur **Generate Installer** pour créer le projet Paquet Builder. XLS Padlock lancera alors Paquet Builder, vous permettant de personnaliser davantage le projet et de construire le programme d'installation final.

**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](#).

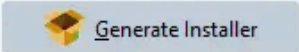
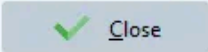
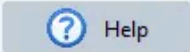
Please fill in the following settings and click **Generate Installer**. A **default project file** to be compiled with Paquet Builder will be created: simply open it in Paquet Builder, modify any settings if needed, and compile it.

Destination Path: c:\program files\

Installer Title:

Your Application Name (used for shortcuts and uninstall display name):

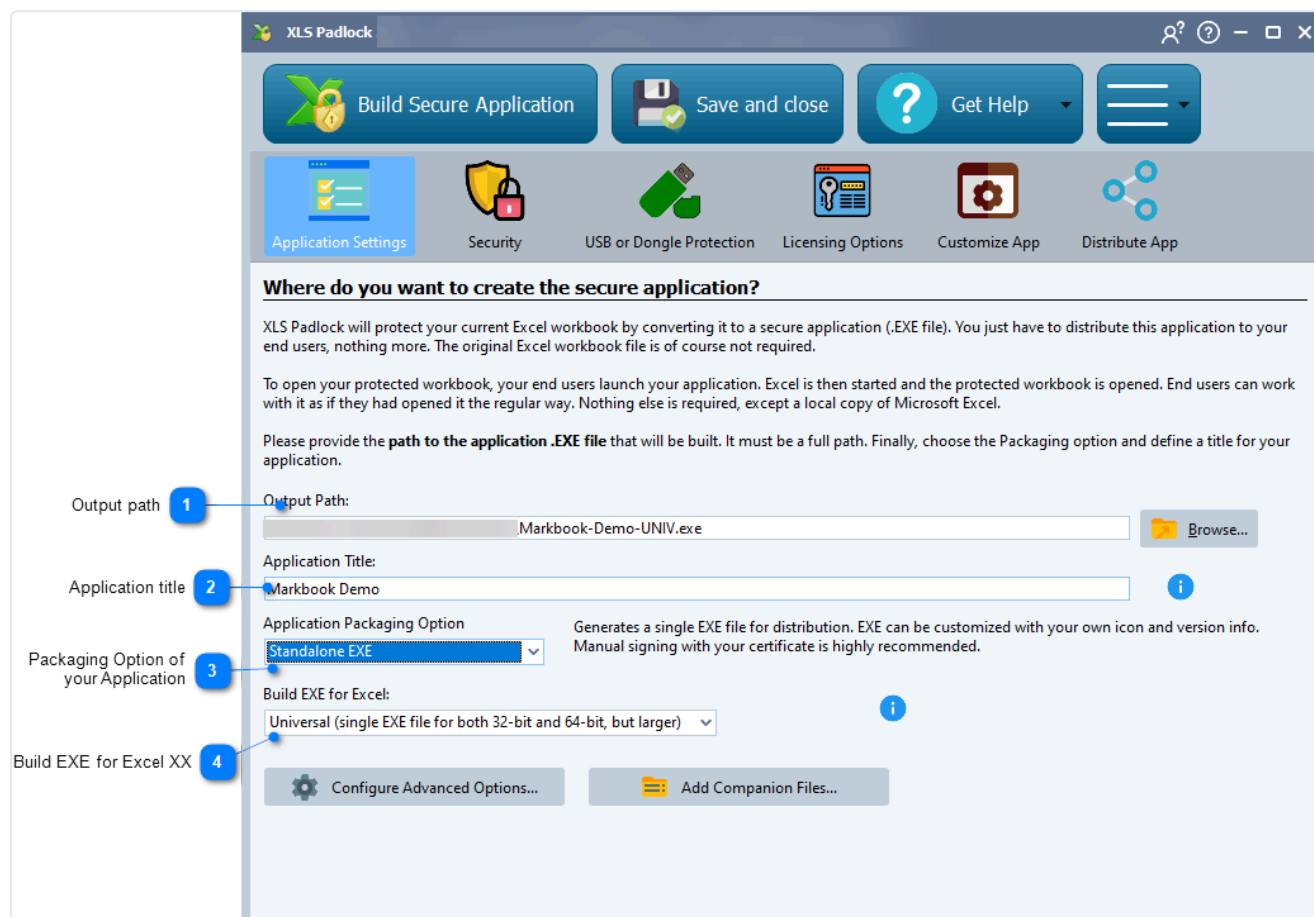
Open the generated project in Paquet Builder

👉 Consultez le [tutoriel complet sur la façon de créer un programme d'installation pour votre classeur Excel](#).

# Compiler l'EXE selon l'architecture d'Excel

Microsoft Excel est disponible en deux architectures : 32 bits et 64 bits. Comme le 64 bits est désormais la norme pour les versions modernes d'Office, la plupart de vos utilisateurs auront besoin d'une application compatible 64 bits. Cependant, certains peuvent encore utiliser la version 32 bits, en particulier sur des systèmes plus anciens. XLS Padlock vous permet de construire pour les deux plateformes.



Cette option n'est disponible que pour les [fichiers exécutables autonomes](#).

👉 Choisissez le type de fichier EXE que XLS Padlock générera :

- **32-bit only** : Génère un fichier EXE compatible uniquement avec les versions 32 bits d'Excel.
- **64-bit only** : Génère un fichier EXE compatible uniquement avec les versions 64 bits d'Excel.
- **Universal** : Crée un seul fichier EXE qui fonctionne à la fois avec les versions 32 bits et 64 bits d'Excel. Cela simplifie la distribution, mais le fichier EXE résultant sera environ deux fois plus volumineux.
- **32-bit and 64-bit** : Génère deux fichiers EXE distincts simultanément. Pour les empêcher de s'écraser mutuellement, XLS Padlock ajoute automatiquement les suffixes "32" et "64" aux noms de fichiers (par exemple, `MyApp32.exe` et `MyApp64.exe`).

### Incompatibilité de version

Si un utilisateur final exécute un EXE avec une architecture Office incompatible (par exemple, un EXE 64 bits sur une installation Office 32 bits), il recevra un message d'erreur lui demandant de vous contacter pour obtenir la bonne version.

### Signature de code recommandée pour le mode Universal

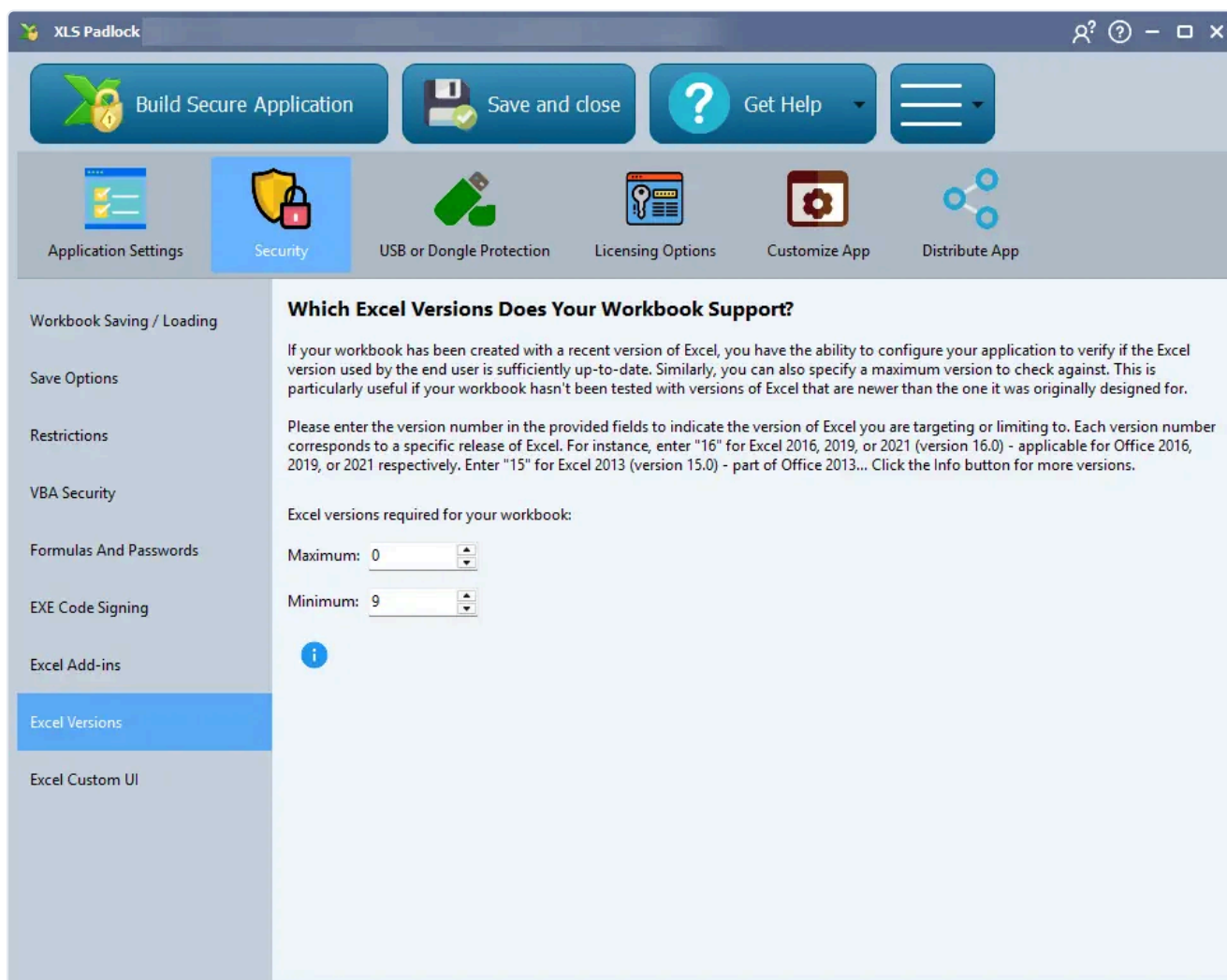
Si vous utilisez le mode Universal, nous vous recommandons fortement de signer numériquement votre application. En effet, un fichier EXE temporaire est créé au moment de l'exécution, ce qui peut parfois être signalé par les logiciels antivirus. Une signature numérique réduit considérablement le risque de faux positifs.

### Info

Par défaut, les [applications distribuées au format bundle](#) sont déjà universelles.

# Versions d'Excel

Si votre classeur nécessite des fonctionnalités d'une version spécifique d'Excel, vous pouvez configurer votre application pour vérifier la version d'Excel de l'utilisateur final au moment de l'exécution.



Vous pouvez définir à la fois une version minimale requise et une version maximale autorisée. Cela est utile pour garantir la compatibilité ou pour empêcher l'utilisation sur des versions d'Excel plus récentes et non testées.

Saisissez le numéro de version correspondant dans les champs :

- **9** : Excel 2000 (version 9.0)
- **10** : Excel 2002 (version 10.0)
- **11** : Excel 2003 (version 11.0)
- **12** : Excel 2007 (version 12.0)
- **14** : Excel 2010 (version 14.0)
- **15** : Excel 2013 (version 15.0)
- **16** : Excel 2016, 2019, 2021 et 365 (version 16.0)

# Mises à jour du classeur

Après avoir distribué votre application de classeur Excel, vous devrez tôt ou tard fournir des mises à jour à vos utilisateurs. XLS Padlock propose plusieurs outils pour simplifier ce processus.

Un point important à considérer pour les mises à jour est la manière dont votre application enregistre les données utilisateur. Vous devez connaître les [différents modes d'enregistrement proposés par XLS Padlock](#) avant de générer votre application.

Si votre classeur est mis à jour souvent et que vos utilisateurs n'ont besoin de modifier que quelques valeurs de cellules, le mode **Save defined cell values only (.XLSCE file)** est le meilleur choix. Ce mode vous permet de mettre à jour le fichier principal du classeur sans affecter les données utilisateur, car XLS Padlock enregistre et restaure [uniquement les valeurs de cellules que vous avez désignées](#).

Lorsque vous mettez à jour votre classeur source, la manière la plus directe de distribuer la mise à jour est de le recompiler et d'envoyer le nouveau fichier EXE à vos clients.

## Flux de travail de mise à jour recommandé

Le comportement du côté de l'utilisateur dépend de ce que vous avez modifié entre les versions :

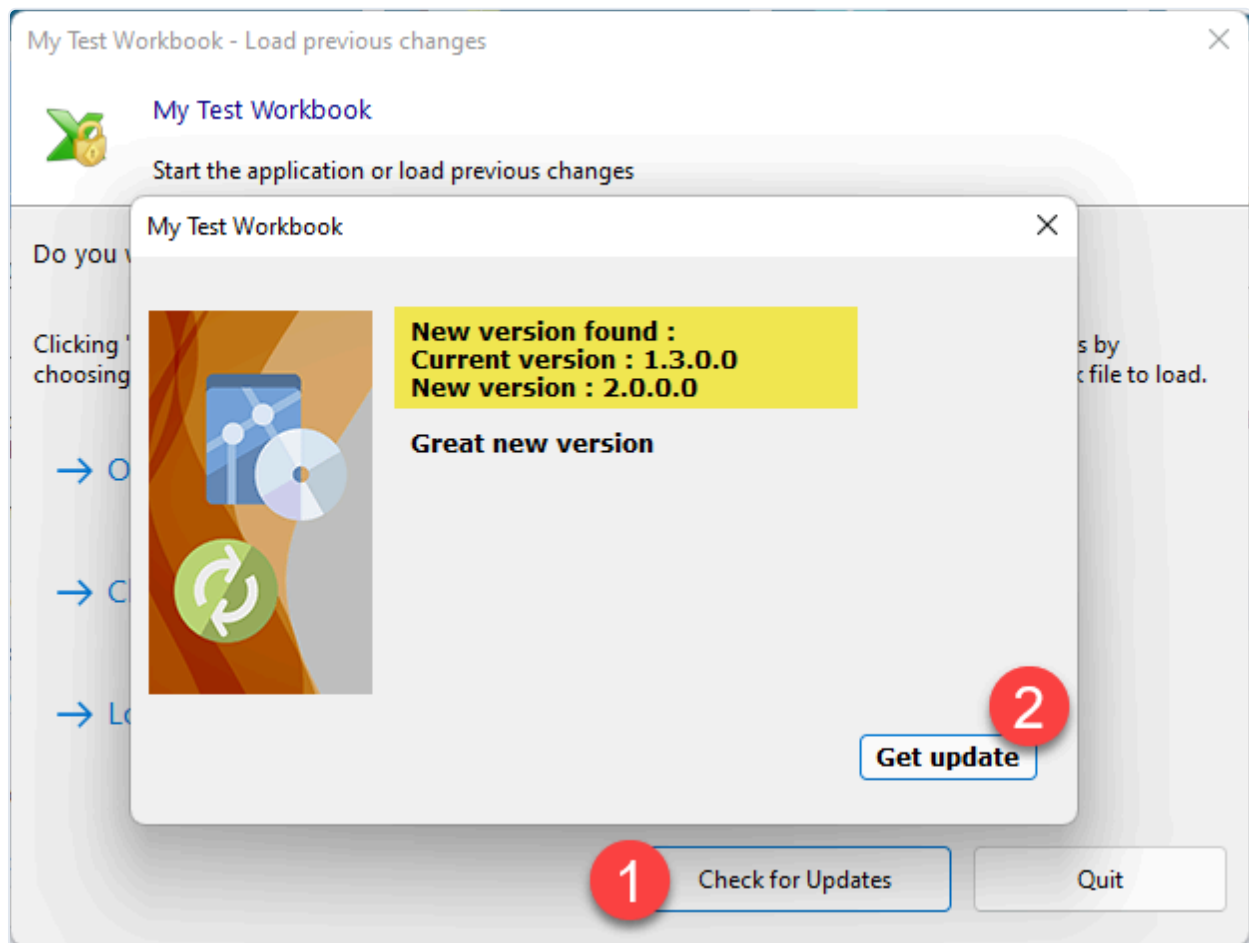
- **Contenu du classeur uniquement** (formules, mise en forme, feuilles, logique VBA) : conservez l'**Application GUID** et la **Secret Key** existants. En mode `.XLSCE`, les utilisateurs verront votre nouvelle logique avec leurs valeurs de cellules enregistrées restaurées. En mode `.XLSC`, les utilisateurs continueront de voir leur dernier instantané enregistré, vos modifications n'apparaîtront que s'ils choisissent manuellement "Original Workbook" depuis l'écran d'accueil.
- **Vous voulez invalider entièrement les anciens fichiers de sauvegarde utilisateur** : générez une nouvelle **Secret Key** dans les paramètres [Activation and Licensing](#), mais laissez l'**Application GUID** inchangé afin que les clés d'activation existantes restent valides. Au premier lancement de l'utilisateur, son ancien fichier de sauvegarde sera détecté comme incompatible et il sera invité, via l'écran d'accueil, à charger le classeur d'origine intégré à la place.
- **Ne changez pas de mode d'enregistrement entre les versions** (`.XLSC` ↔ `.XLSCE`). Les fichiers de sauvegarde existants issus du mode précédent seront ignorés, et toute donnée que vos utilisateurs n'avaient pas encore ressaisie sera perdue. Choisissez un mode d'enregistrement au début du projet et conservez-le pendant toute la durée de vie de l'application.

### CONSEIL

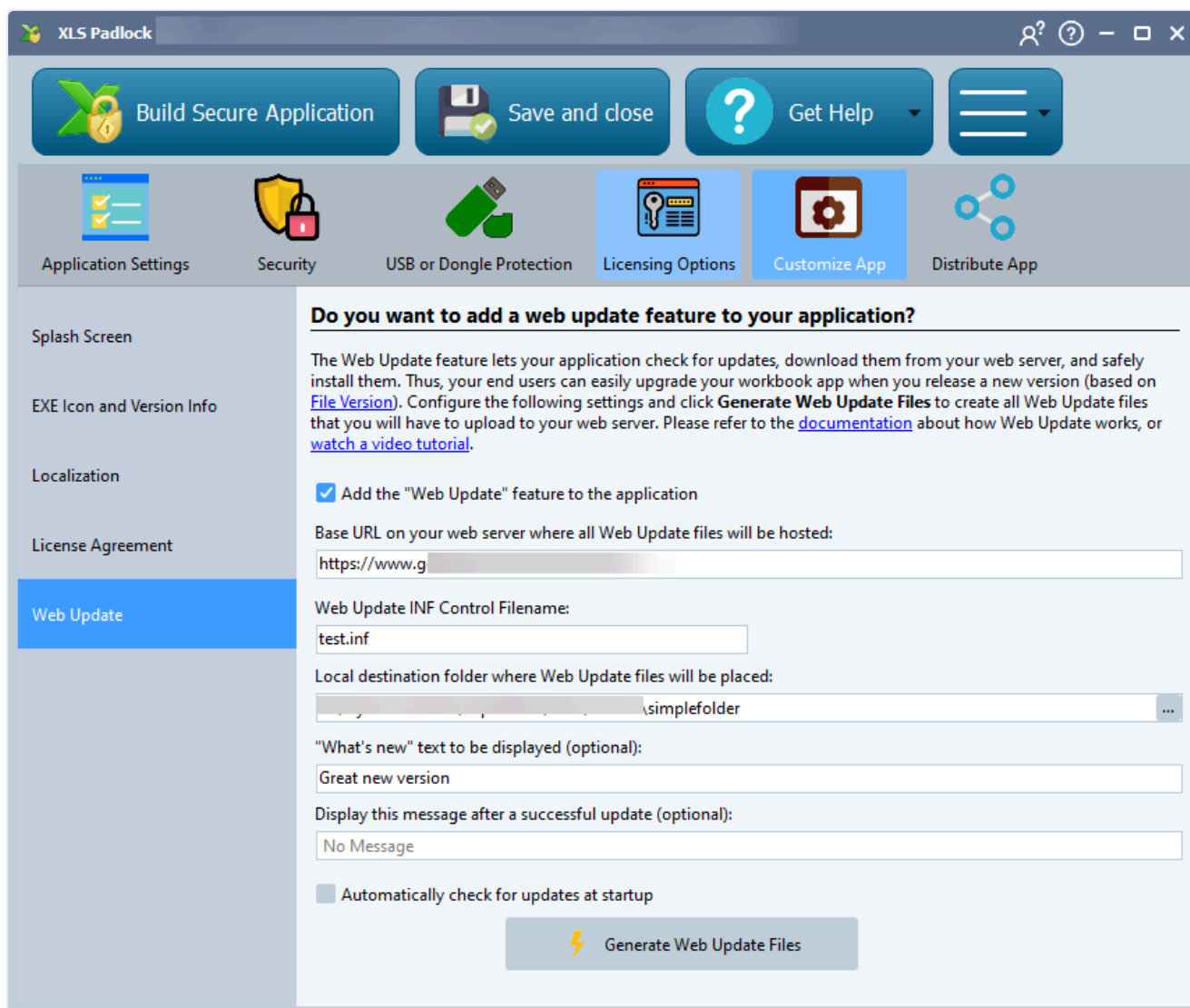
Le bouton [Original Workbook](#) sur l'écran d'accueil sert de filet de sécurité à l'utilisateur lorsqu'un ancien fichier de sauvegarde ne parvient pas à se charger. Même si vous l'avez masqué, XLS Padlock le réactivera automatiquement pour la nouvelle tentative, afin que les utilisateurs ne soient jamais bloqués.

👉 Cependant, télécharger et remplacer manuellement le fichier EXE peut être fastidieux pour les utilisateurs finaux. Pour simplifier cela, XLS Padlock propose une fonctionnalité de mise à jour

automatique par le web. Vous pouvez configurer votre application pour qu'elle vérifie en ligne la disponibilité de mises à jour et avertisse les utilisateurs lorsqu'une nouvelle version est disponible :



Cette fonctionnalité peut être configurée rapidement et directement dans XLS Padlock :



👉 Pour mettre en place votre propre système de mise à jour, vous aurez besoin d'un serveur web ou d'un espace d'hébergement où vous pourrez placer des fichiers en téléchargement direct. Suivez ensuite les instructions de la page [comment configurer les mises à jour automatiques par le web](#).

### Tutoriel vidéo

Regardez notre tutoriel vidéo sur [comment configurer les mises à jour automatiques par le web pour vos classeurs Excel](#).

# Mises à jour web automatiques

XLS Padlock inclut une fonctionnalité de mise à jour par le web qui peut télécharger et installer automatiquement de nouvelles versions de votre application. Cela permet à vos utilisateurs finaux de mettre facilement à niveau votre application de classeur chaque fois que vous publiez une nouvelle version.

Aucun logiciel tiers n'est requis, mais **vous devez disposer d'un serveur web ou d'un espace d'hébergement web** pour héberger les fichiers de mise à jour.

## Tutoriel vidéo

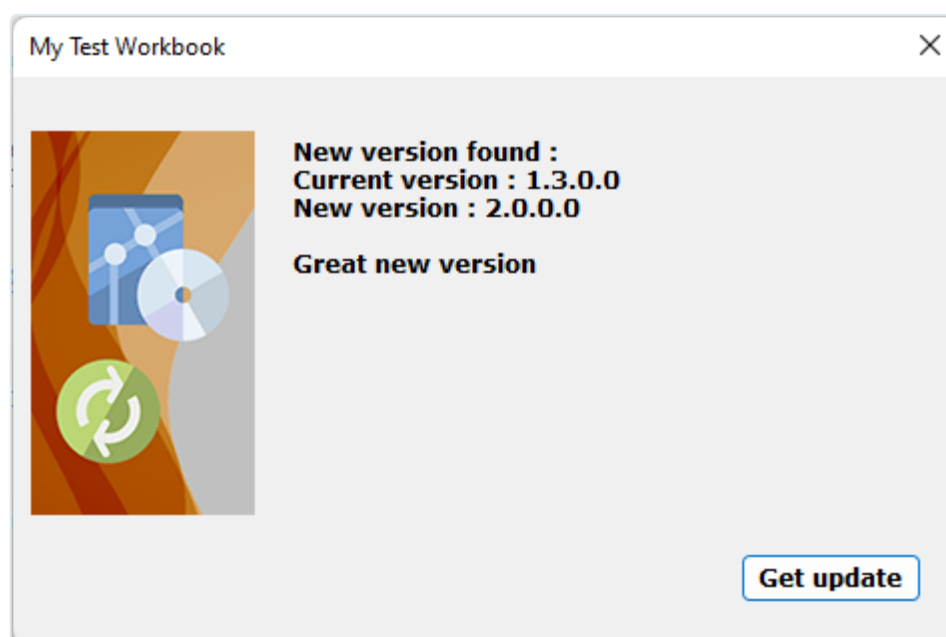
Regardez un [tutoriel vidéo](#) sur la configuration des mises à jour automatiques par le web pour les classeurs Excel.

## Comment fonctionne la mise à jour par le web

L'application télécharge d'abord un petit fichier de contrôle ( `.inf` ) depuis votre serveur. Ce fichier indique à l'application si une nouvelle version est disponible. Si tel est le cas, l'application télécharge les fichiers de mise à jour nécessaires ( `.cab` ), se ferme, installe la mise à jour et redémarre.

## Vérification des mises à jour

Vous pouvez configurer votre application pour qu'elle **vérifie automatiquement les mises à jour au démarrage** ou ajouter un bouton "**Check for Updates**" à l'[écran d'accueil](#). Lorsqu'une nouvelle version est détectée, les utilisateurs sont guidés à travers un assistant de mise à jour par le web.



Les utilisateurs peuvent également lancer une mise à jour via le [commutateur de ligne de commande](#) `webupdate` .

## Comment une nouvelle version est détectée

Le numéro de **File Version**, spécifié sur la page [EXE Version Info](#) de XLS Padlock, est utilisé pour déterminer si une mise à jour est nécessaire. Vous devez augmenter ce numéro de version à chaque fois que vous publiez une mise à jour.

## Configuration

- **Base URL** : l'URL sur votre serveur web où les fichiers de mise à jour par le web seront hébergés (par exemple, `https://www.yourwebsite.com/myfolder` ). HTTPS est recommandé.
- **Web Update INF Control Filename** : le nom du fichier de contrôle qui contient les informations de mise à jour.
- **Local Destination Folder** : un dossier local sur votre ordinateur où XLS Padlock générera les fichiers de mise à jour. Vous devez ensuite téléverser l'intégralité du contenu de ce dossier vers votre serveur web.

### Avertissement

Le dossier de destination doit être vide. XLS Padlock demandera à en supprimer le contenu s'il ne l'est pas.

- **Texte "What's new" (facultatif)** : texte à afficher dans l'assistant de mise à jour, informant les utilisateurs des changements de la nouvelle version.

## Génération des fichiers de mise à jour par le web

Cliquer sur **Generate Web Update Files** dans XLS Padlock créera les fichiers nécessaires (un fichier de contrôle `.inf` et un fichier compressé `.cab` ) dans le dossier local que vous avez spécifié. Vous pourrez ensuite téléverser ces fichiers vers votre serveur web.

### Privilèges administrateur

Si l'application est installée dans un dossier Windows restreint comme `Program Files` , des privilèges administrateur seront requis pour installer la mise à jour, ce qui déclenchera une invite UAC.

## Dépannage

Pour dépanner la fonctionnalité de mise à jour par le web, activez l'option **"Enable WebUpdate Log"** dans les [Options avancées](#). Cela créera un fichier `WUPDATE.LOG` dans le dossier Documents de l'utilisateur, contenant des informations détaillées sur le processus de mise à jour.

# Migrer les données utilisateur entre les mises à jour

Lorsque vous mettez à jour votre classeur protégé et distribuez un nouvel EXE, vous pouvez rencontrer des difficultés pour garantir le transfert correct des données de vos utilisateurs, surtout si vous utilisez le [mode Full Save](#). Ce mode crée un **instantané complet du classeur** au moment de l'enregistrement. Par conséquent, lorsqu'un utilisateur ouvre son ancien fichier de sauvegarde ( `.xlsb` ) avec la nouvelle application, il verra ses anciennes données dans l'ancienne structure du classeur et ne bénéficiera pas de vos dernières mises à jour (telles que de nouvelles fonctionnalités ou des corrections de bogues dans votre code VBA).

Cette rubrique explique comment utiliser un système d'exportation/importation basé sur VBA pour aider vos utilisateurs à migrer leurs données d'une ancienne version de votre application vers une nouvelle.

## Avertissement

La solution décrite ici dépend du [mode d'enregistrement que vous avez sélectionné pour votre projet XLS Padlock](#). Si vous utilisez le mode d'enregistrement **Cell Values**, les données utilisateur sont automatiquement chargées dans la structure de votre classeur mis à jour, et ce processus manuel n'est pas nécessaire.

## Alternative : mode d'enregistrement Cell Values

Comme mentionné ci-dessus, le [mode d'enregistrement Cell Values](#) est le moyen le plus simple de gérer les mises à jour. Cependant, ce mode est conçu pour n'enregistrer que les valeurs de cellules spécifiques. Si vous avez besoin de transférer d'autres types de données, comme l'état de variables ou d'objets VBA, vous devrez utiliser une solution VBA personnalisée comme celle décrite dans cette rubrique.

## ## Exemple de code

Ce processus comporte deux étapes principales : exporter les données depuis l'ancienne version et les importer dans la nouvelle.

### 1. Exporter les données utilisateur vers un fichier Excel

La macro ci-dessous génère un fichier Excel `.xlsx` normal contenant les données de l'utilisateur. Vous devez inclure cette macro dans votre ancienne application, liée à un bouton, afin que les utilisateurs puissent exporter leurs données avant de mettre à niveau.

#### REMARQUE

Vous devrez remplacer les noms de feuilles et les plages de cellules dans la macro ci-dessous pour correspondre à la structure de votre classeur.

```
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. Téléverser les données vers le nouveau fichier EXE.

L'utilisateur doit ouvrir le nouvel EXE et exécuter la 3e macro ci-dessous pour téléverser les données (liez la macro à un bouton). Une fois que l'utilisateur exécute la macro, il sera invité à sélectionner le fichier (1re macro), et les données seront copiées (2e macro). La troisième macro exécute les deux macros, et c'est cette macro qui doit être liée au bouton. Là encore, vous devrez modifier les cellules et les noms d'onglets dans la deuxième macro, et vous pouvez également modifier le titre dans la première 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
```

#### REMARQUE

Afin d'enregistrer des classeurs normaux, vous pouvez aussi consulter :  
[Chargement/Enregistrement de classeurs via l'assistant VBA SetOption](#)

👉 Voir aussi :

- [Mode d'enregistrement que vous avez sélectionné pour votre projet XLS Padlock](#)

# Enregistrer et restaurer les réglages via des modèles

---

XLS Padlock vous permet d'enregistrer les paramètres actuels de votre projet dans un fichier modèle. Il s'agit d'une fonctionnalité puissante qui peut vous faire gagner un temps considérable, en particulier si vous créez fréquemment de nouvelles applications de classeur avec des configurations similaires.

## Enregistrer les paramètres du projet dans un modèle

---

Une fois que vous avez configuré toutes les options souhaitées pour votre projet (telles que les paramètres de sécurité, la personnalisation et les options de sortie), vous pouvez enregistrer cette configuration en tant que modèle.

Cliquez sur le bouton



App Menu puis sur le menu **Save Current Settings As Template**. Vous serez invité à choisir un emplacement et un nom pour votre fichier modèle (qui portera l'extension `.XPLP`).

### Avertissement

Ne choisissez pas votre fichier de projet XLS Padlock existant.

## Restaurer les paramètres à partir d'un modèle

---

Lorsque vous démarrez un nouveau projet ou que vous souhaitez appliquer une configuration standard à un projet existant, vous pouvez facilement restaurer les paramètres à partir d'un modèle enregistré précédemment.

Cliquez sur le bouton



App Menu, puis sur le menu **Restore Project from Template**, accédez à votre fichier modèle `.XPLP` et sélectionnez-le. Tous les paramètres stockés dans ce modèle seront instantanément appliqués à votre projet actuel.

**REMARQUE**

Le chemin de sortie de l'application n'est pas stocké dans les modèles.

# Erreur de registre ou EREGISTRYEXCEPTION

---

Cette erreur indique souvent un conflit avec un logiciel de sécurité ou un programme antivirus. Pour la résoudre, activez l'option « **Use another registry key for storing activation data** » dans les [Advanced Options](#).

# Violation d'accès à l'adresse

---

Cette erreur peut se produire lors de l'utilisation de la fonctionnalité « Lock VBA Project (simple VBA protection) ». Pour la résoudre, supprimez le mot de passe de votre projet VBA avant de compiler le classeur avec XLS Padlock. Le mot de passe n'est pas nécessaire, car XLS Padlock verrouillera le projet automatiquement sans en demander un.

Vous pouvez également essayer d'utiliser un mot de passe plus court ou passer à l'option [Prevent access to VBA editor](#).

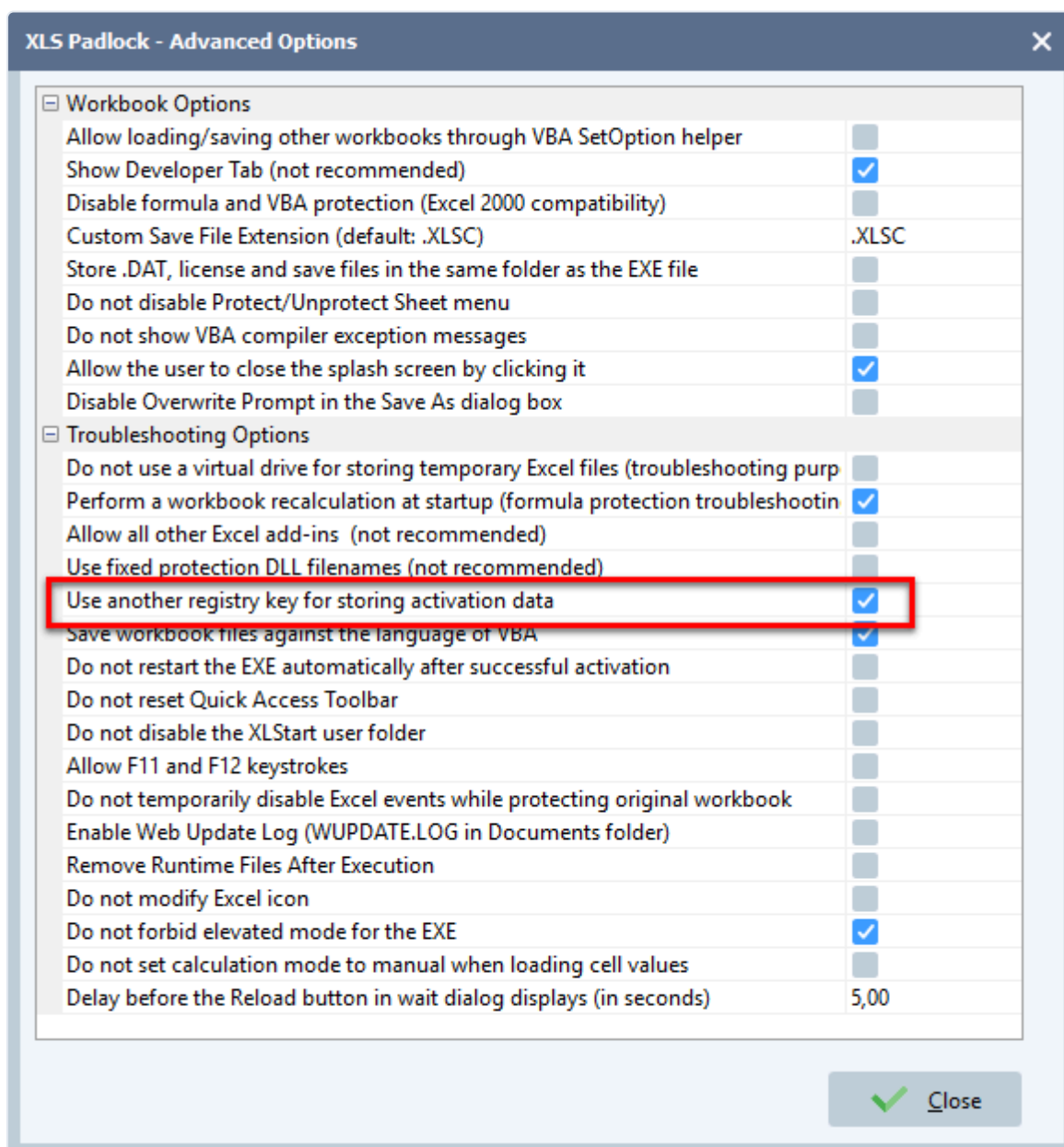
# Erreur « Failed to set data for 'Data' »

Si votre utilisateur final reçoit le message d'erreur suivant lors de la saisie d'une clé d'activation, cela indique généralement un conflit avec un logiciel de sécurité sur son système.

## Error setting data in registry, Failed to set data for 'Data'

Cette erreur se produit parce qu'un autre programme, généralement un antivirus ou une suite de sécurité, empêche l'application d'écrire ses données d'activation dans le Registre Windows.

Pour résoudre ce problème, vous pouvez configurer votre projet XLS Padlock afin qu'il utilise une autre clé de registre. Activez l'option avancée **Use another registry key for storing activation data** (utiliser une autre clé de registre pour stocker les données d'activation), comme illustré ci-dessous, puis recompilez votre application.



# Pourquoi l'EXE est-il si volumineux

---

Le fichier `.exe` compilé est nettement plus volumineux que le classeur d'origine, car il s'agit d'une application autonome qui regroupe plusieurs composants essentiels :

- Un chargeur sécurisé compatible avec différentes versions d'Excel et de Windows.
- Un moteur d'exécution complet prenant en charge l'Unicode.
- Le compilateur et l'interpréteur VBA propriétaires.
- Plusieurs couches de protection anti-piratage et de protection du code.

Pour réduire la taille du fichier EXE :

- **Use UPX Compression** (utiliser la compression UPX) : UPX est un compresseur d'EXE gratuit et open source, disponible sur [upx.github.io](https://upx.github.io). Compresser l'EXE peut réduire considérablement sa taille. Nous vous recommandons toutefois vivement de signer numériquement tout fichier EXE compressé, car certains antivirus pourraient sinon déclencher des faux positifs.
- **Disable Formula Protection** (désactiver la protection des formules) : si votre application ne repose pas sur des formules sensibles, vous pouvez désactiver la protection des formules afin de réduire légèrement la taille finale du fichier EXE.

# Restaurer le fichier XLS depuis l'EXE

---

Non, il n'est pas possible de restaurer le fichier Excel d'origine à partir d'un .EXE compilé si la fonction « Save As » a été désactivée.

Si vous avez besoin de pouvoir récupérer le classeur d'origine, vous devez activer la fonction « Save As » lors de la compilation. Cela vous permet d'utiliser la fonctionnalité de [déchiffrement des fichiers enregistrés](#).

# Liens d'assistance

---

Si vous avez des questions ou des problèmes avec XLS Padlock, vous pouvez nous joindre via les canaux suivants :

- **Forum des utilisateurs** : <https://www.gdgsoft.info> (Choisissez la catégorie « XLS Padlock »)
- **Support par e-mail** : [info@xlspadlock.com](mailto:info@xlspadlock.com)
- **Pour le support du dongle Enky SL/LC** : [info@hs-securityware.com](mailto:info@hs-securityware.com)

Lorsque vous demandez de l'assistance, veuillez indiquer votre version d'Excel (par exemple, Excel 365, Excel 2019) et son architecture (32 bits ou 64 bits) afin de nous aider à résoudre votre problème plus rapidement.

---

**Suivez-nous sur X** : <https://x.com/gdgsoft>

**Nos autres produits sont disponibles sur** : <https://www.gdgsoft.com>