

XLS PADLOCK

Guía del usuario de XLS Padlock

Protección, licencias y distribución de libros de Excel

Version 2026.0

G.D.G. Software

www.xlspadlock.com

Copyright © G.D.G. Software 2018-2026. Todos los derechos reservados.

Tabla de contenidos

- 01 Acerca de esta guía

- 02 Introducción a XLS Padlock

- 03 Descargar e instalar

- 04 Configuración de la aplicación

- 05 XLS Padlock Manager

- 06 Cómo proteger un libro de Excel

- 07 Seguridad

- 08 Opciones de nube e IA

- 09 Mejorar la protección

- 10 Control de acceso al libro

- 11 Restricciones del libro de Excel

- 12 Protección por contraseña de las hojas

- 13 Combinar la protección de Excel y XLS Padlock

- 14 Protección de fórmulas: Excel frente a XLS Padlock

- 15 Proteger fórmulas con XLS Padlock

- 16 Marcar celdas para proteger

- 17 Método de protección de fórmulas

18 Desactivar la protección de fórmulas

19 Acerca del VBA Compiler integrado

20 Protección real del código VBA

21 Referencia de sintaxis

22 Recetario de la API de VBA

23 Escribir y compilar VBA seguro

24 Invocar VBA compilado en tiempo de ejecución

25 Acceder a los objetos de Excel

26 Pasar matrices

27 Pasar más parámetros

28 Gestionar errores

29 Error OLE 800A03EC

30 Protección del código VBA

31 Ocultar y bloquear el código VBA

32 Bloquear el proyecto VBA

33 Prohibir el acceso al VBE

34 Guía de activación y licencias

35 Claves de activación

36 Configurar claves de activación

37 Claves de activación bloqueadas al hardware

38 Activación en línea

39 Validación en línea

40 Editor del formulario de registro

41 Mostrar el acuerdo de licencia

42 Generador de claves (portátil y servidor)

43 SDK del generador de claves

44 Generador de claves autónomo

45 Restricciones de las claves

46 Protección por USB o mochila (dongle)

47 Desactivación

48 Crear libros de prueba

49 Cerrar tras un tiempo determinado

50 Comprobar el estado de prueba

51 Días de prueba restantes

52 Opciones de guardado del libro

53 Guardado y carga

54 Modo de guardado: completo o valores de celdas

55 Definir celdas para guardar y restaurar

- 56 Restaurar y guardar con VBA

- 57 Acceder al libro seguro y a los archivos complementarios

- 58 Añadir archivos complementarios

- 59 Carpeta de almacenamiento de guardados

- 60 Almacenar los cambios dentro del EXE

- 61 Restringir la carga y el guardado

- 62 Abrir y descifrar los guardados

- 63 Bloquear los guardados a un equipo

- 64 Referencias externas e hipervínculos

- 65 Obtener la ruta junto al libro compilado

- 66 Personalizar la aplicación

- 67 Empaquetado de la aplicación

- 68 Pantalla de bienvenida

- 69 Cambiar el icono del EXE

- 70 Ventana de Excel al inicio

- 71 Ejecutar como aplicación solo VBA

- 72 Información de versión del EXE

- 73 Parámetros de línea de comandos

- 74 Ocultar el cuadro de diálogo de carga

- [75 Localización y traducción](#)

- [76 Opciones avanzadas](#)

- [77 Ignorar errores de procesamiento](#)

- [78 Desactivar la información de depuración](#)

- [79 Interfaz personalizada de Excel](#)

- [80 Complementos de Excel](#)

- [81 Distribuir la aplicación](#)

- [82 Distribuir un libro protegido](#)

- [83 Firmar digitalmente el EXE](#)

- [84 Crear un instalador](#)

- [85 Compilar el EXE según la arquitectura de Excel](#)

- [86 Versiones de Excel](#)

- [87 Actualizaciones del libro](#)

- [88 Actualizaciones web automáticas](#)

- [89 Migrar los datos de usuario entre actualizaciones](#)

- [90 Guardar y restaurar la configuración con plantillas](#)

- [91 Error de registro o EREGISTRYEXCEPTION](#)

- [92 Violación de acceso en la dirección](#)

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

94 Por qué el EXE es tan grande

95 Restaurar el archivo XLS desde el EXE

96 Enlaces de soporte

Acerca de esta guía

Copyright © G.D.G. Software 2013-2026

Todos los derechos reservados. Ninguna parte de esta obra puede reproducirse de ninguna forma ni por ningún medio, gráfico, electrónico o mecánico, incluidas la fotocopia, la grabación, el registro en cinta o los sistemas de almacenamiento y recuperación de información, sin el permiso por escrito del editor.

Se concede a los clientes con licencia de XLS Padlock permiso para imprimir esta guía para uso privado o educativo.

Microsoft Excel® y Office® son marcas comerciales registradas de Microsoft Corporation.

Cualquier producto mencionado en este documento puede ser marca comercial o marca comercial registrada de sus respectivos propietarios. El editor y el autor no reclaman ningún derecho sobre estas marcas comerciales.

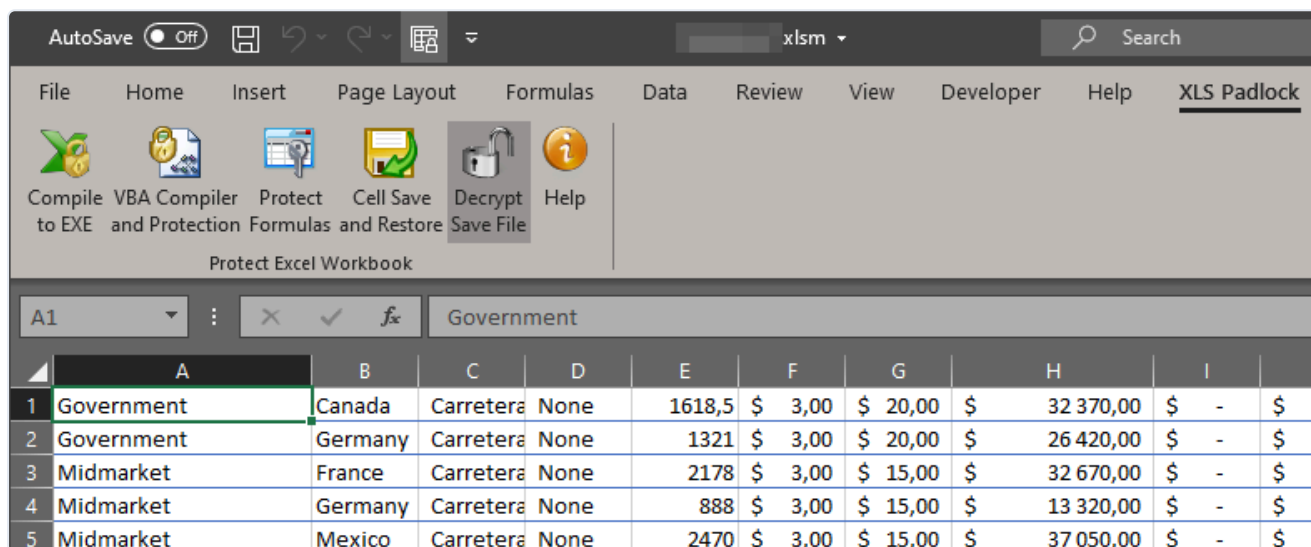
Aunque se han tomado todas las precauciones al preparar este documento, el editor y el autor no asumen ninguna responsabilidad por errores u omisiones, ni por daños resultantes del uso de la información aquí contenida o del uso de los programas y el código fuente que la acompañan. En ningún caso el editor o el autor serán responsables de ninguna pérdida de beneficios ni de ningún otro daño comercial causado o presuntamente causado directa o indirectamente por este documento.

Primera edición: enero de 2013. Actualizado: junio de 2026.

Version 2026.0, [Leer el registro de cambios](#)

Introducción a XLS Padlock

XLS Padlock es **una potente solución de protección y licenciamiento para Microsoft Excel**, diseñada para proteger sus libros contra copias. Funciona como un compilador, lo que le permite convertir sus libros de Excel en aplicaciones autónomas y seguras.



Esto le permite distribuir sus archivos de Excel de forma segura, controlar el acceso de los usuarios e impedir la copia de hojas de cálculo. Se pueden restringir funciones clave de Excel, entre ellas:

- El menú contextual de la hoja de cálculo (clic derecho)
- La funcionalidad de copiar y pegar
- La barra de fórmulas
- Guardar e imprimir
- El acceso al proyecto VBA

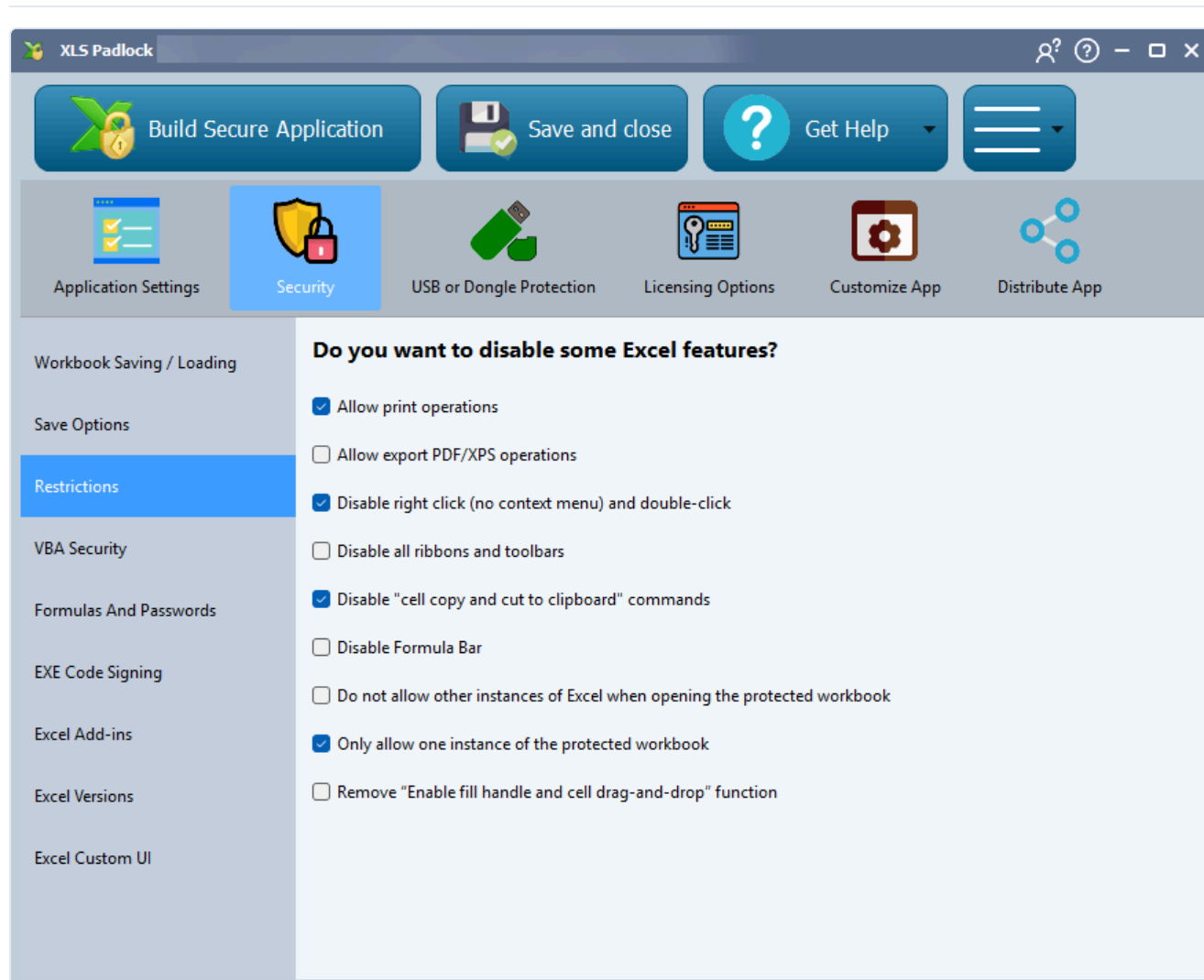
XLS Padlock ofrece múltiples formas de proteger sus archivos, como **bloquearlos a una memoria USB específica**, utilizar una **sólida protección por dongle** o emitir **claves de activación bloqueadas por hardware**. También obtiene un control granular sobre la protección de fórmulas, lo que permite a los usuarios usar sus fórmulas sin poder verlas ni copiarlas.

Protección avanzada del código VBA

Con su VBA Compiler integrado, XLS Padlock le permite **compilar el código VBA sensible en byte-code seguro, haciéndolo inaccesible para los usuarios finales**. Sus macros VBA quedan realmente protegidas porque el código fuente original se elimina.

El compilador **no es un simple ofuscador**; transforma el código VBA en código binario y lo almacena de forma segura dentro del EXE de la aplicación. Para completar la protección, puede bloquear su proyecto VBA, **dejando inútiles las herramientas para descifrar contraseñas**. [Más información sobre la protección del código VBA.](#)

Opciones de seguridad completas



Las aplicaciones se pueden configurar para que **caduquen después de un número determinado de días**, un número específico de usos o en una fecha fija. Se puede mostrar una pantalla de aviso personalizable para las versiones de prueba. La **activación en línea** automatiza el proceso de licenciamiento, lo que le otorga **control remoto** sobre quién puede acceder a sus libros.

Las aplicaciones de libros protegidos son autónomas y solo requieren Microsoft Excel para funcionar. Todas las funciones de Excel son totalmente compatibles.

👉 También puede personalizar y aplicar su marca a sus aplicaciones:

- Añadir un icono personalizado e información de copyright.
- Traducir todo el texto visible para el usuario a cualquier idioma.
- Mostrar un acuerdo de licencia (EULA) o una pantalla de bienvenida al inicio.
- Firmar digitalmente sus libros compilados con Authenticode.
- Crear un instalador profesional para la distribución.
- Notificar a los usuarios de nuevas versiones con la función integrada de actualización web.

Compatible con Office 365 y Excel 2024, 2021, 2019, 2016, 2013, 2010, 2007 (SP3), 2002, tanto en versiones de 32 bits como de 64 bits.

Advertencia

Se requiere una versión de 64 bits de Windows para crear aplicaciones de 64 bits.

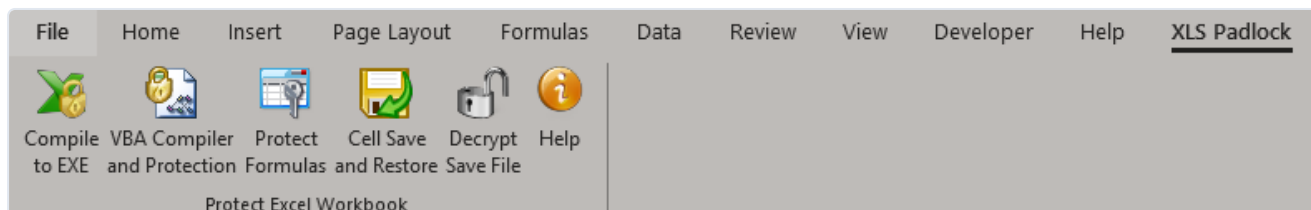
👉 Explore ahora:

- [Descargar e instalar XLS Padlock](#)
- [Cómo proteger un libro de Excel](#)
- [Sitio web de XLS Padlock](#)
- [Novedades](#)

Descargar e instalar

En nuestro sitio web, en <https://www.xlspadlock.com/es/descargar>, debe seleccionar el instalador que coincida con su versión de Microsoft Excel (32 bits o 64 bits). El instalador verificará que esté instalando la versión correcta.

Tras una instalación correcta, **XLS Padlock se integra directamente en Excel**, apareciendo como una nueva pestaña en la cinta de opciones. Para usar XLS Padlock, primero debe abrir su libro de Excel.



También se crean dos accesos directos en su escritorio: uno para esta guía del usuario y otro para el XLS Padlock Manager.

- [XLS Padlock Manager](#)
- [Cómo proteger un libro de Excel](#)

Configuración de la aplicación

La página Application Settings contiene los ajustes obligatorios de su proyecto.

Ajustes principales

Output Path

👉 Define dónde se creará el archivo EXE final.

XLS Padlock compila su libro en un único archivo ejecutable (EXE). En el campo "Output Path" debe especificar la ruta completa y el nombre de archivo donde desea que se cree su aplicación.

CONSEJO

No olvide incluir la extensión `.exe` en el nombre de archivo.

También se aceptan rutas relativas; serán relativas a la carpeta que contiene el archivo de su libro de origen. De forma predeterminada, XLS Padlock sugiere una ruta en la misma carpeta que su libro de origen, pero puede cambiarla a cualquier ubicación.

Application Title

👉 Establece el título de la ventana principal de su aplicación.

Puede establecer un título personalizado para su aplicación. Este título se muestra en la barra de título de la ventana, reemplazando el texto predeterminado "Microsoft Excel".

Si desea mostrar dinámicamente el nombre de archivo de un archivo guardado cargado, puede utilizar los siguientes marcadores de posición en el título:

- `%SAVEFILENAME%` : se reemplaza únicamente por el nombre de archivo (por ejemplo, `MyData.xlsx`).
- `%SAVEFULLNAME%` : se reemplaza por la ruta completa al archivo guardado (por ejemplo, `C:\Users\Me\Documents\MyData.xlsx`).

Por ejemplo, un título establecido en `My Application, %SAVEFILENAME%` se mostraría como "My Application, MyData.xlsx" cuando el usuario tiene ese archivo abierto.

NOTA

Si se carga el libro original sin guardar, los marcadores de posición estarán vacíos.

Application Packaging Option

Consulte [Packaging Option](#): elija entre un EXE independiente o un paquete para la distribución.

Build EXE for Excel

Consulte [Build EXE for Excel](#): especifique si desea compilar para 32 bits, 64 bits o compatibilidad universal.

Configuración adicional

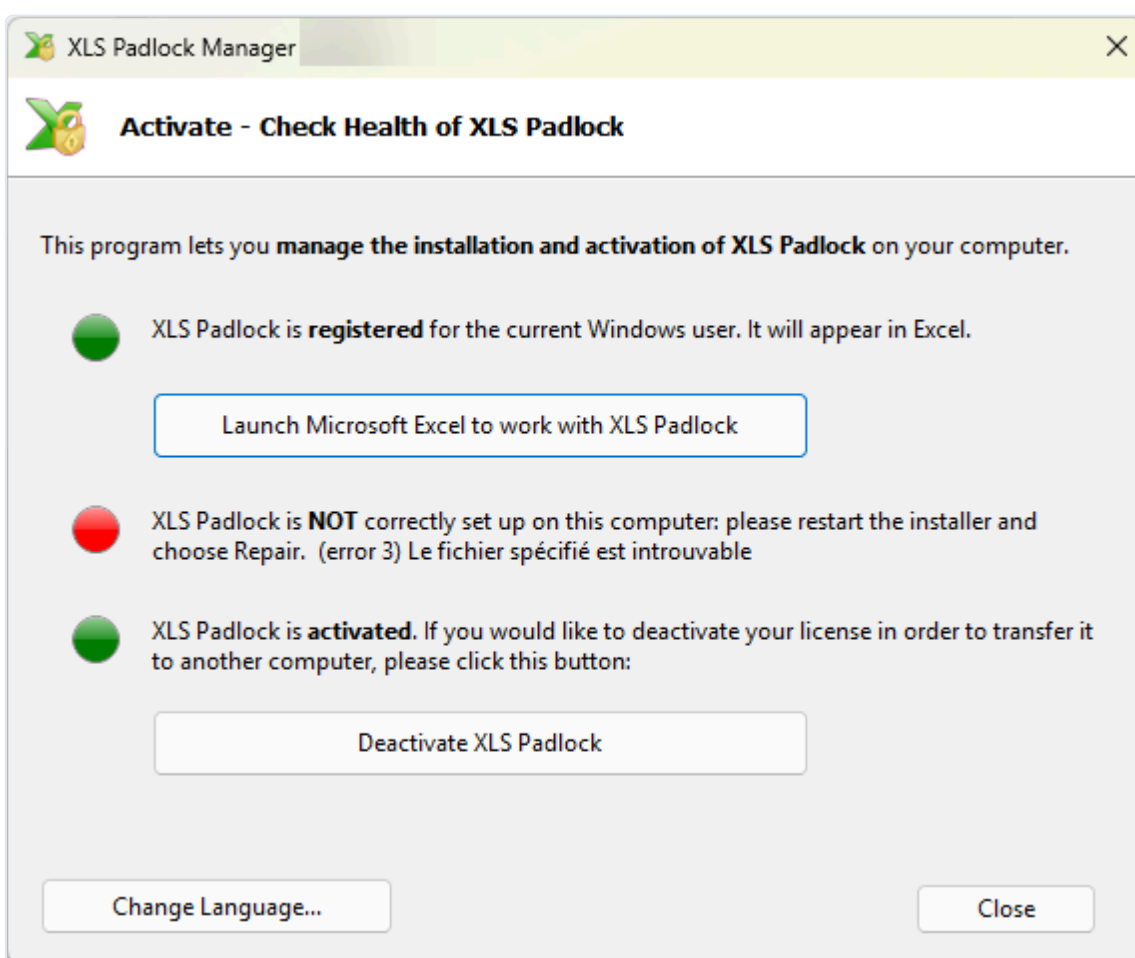
- [Configure Advanced Options](#)
- [Add Companion Files](#)

XLS Padlock Manager

XLS Padlock Manager es una aplicación independiente que le permite **comprobar si XLS Padlock está configurado y registrado correctamente para el usuario actual de Windows**.

XLS Padlock **debe estar registrado para cada usuario de Windows que desee utilizarlo**. Cuando instala XLS Padlock en un equipo, normalmente solo queda registrado para la cuenta de administrador que realizó la instalación. Las demás cuentas de usuario deben utilizar el XLS Padlock Manager para registrar el software.

- Para registrar XLS Padlock para su cuenta de usuario de Windows, inicie "XLS Padlock for Excel, Manager" desde el escritorio de Windows. Aparecerá la siguiente ventana:



- Si todos los indicadores están en verde, su instalación es correcta.
- Si el primer indicador está en rojo, haga clic en "**Enable XLS Padlock for the current Windows user**". XLS Padlock quedará entonces registrado y podrá empezar a utilizarlo. Esta acción no requiere derechos de administrador.

**NOTA**

Si dispone de una licencia de XLS Padlock, también puede utilizar el manager para activarla.

 [Cómo proteger un libro de Excel](#)

Cómo proteger un libro de Excel

Tutorial en vídeo disponible

Consulte estos tutoriales en vídeo sobre XLS Padlock de la mano de un experto en Excel:

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

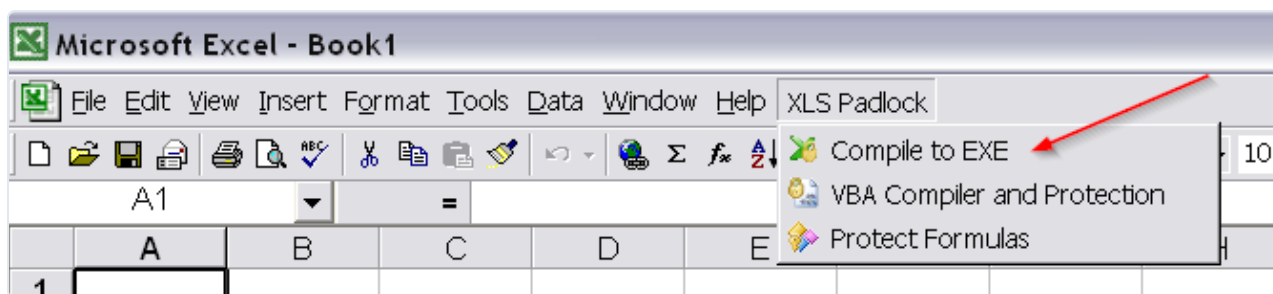
1\ Abra y guarde su libro

Comience abriendo en Excel el libro que desea proteger. Asegúrese de que todos los cambios estén guardados antes de continuar.

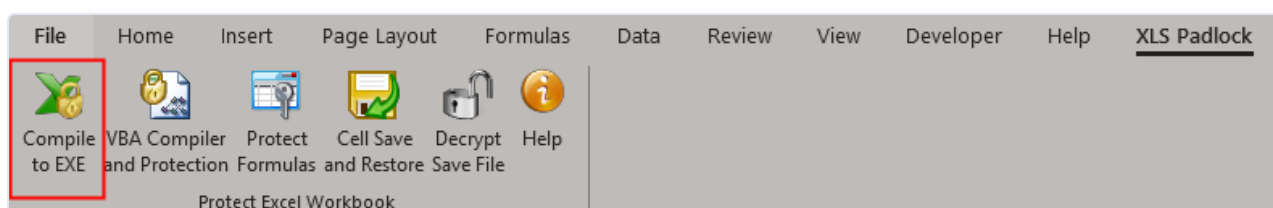
2. Configure las opciones de protección

Para empezar, abra la interfaz de XLS Padlock desde la cinta de opciones de Excel:

- **Antes de Excel 2007:** utilice el menú "XLS Padlock".



- **Excel 2007 y posteriores:** haga clic en "Secure Compile to Exe" en la pestaña "XLS Padlock" de la cinta de opciones.



NOTA

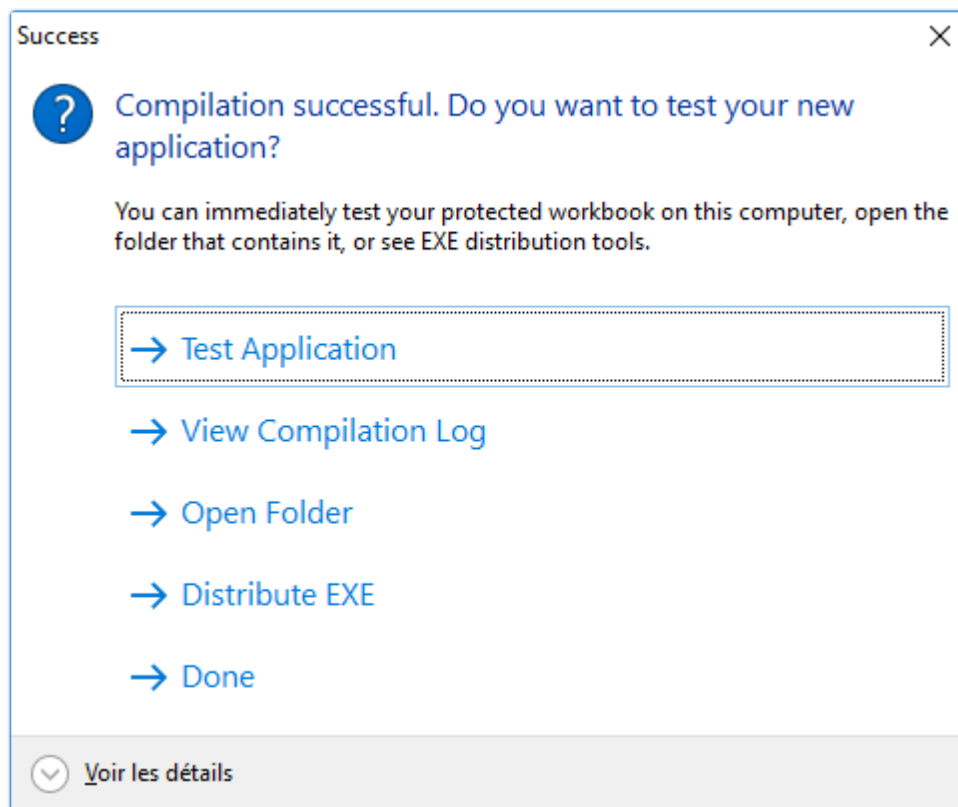
Guarde siempre su libro de Excel antes de abrir la interfaz de XLS Padlock para asegurarse de que todos los cambios recientes se incluyan en la compilación.

3. Compile su aplicación

Tras configurar los ajustes de su aplicación, haga clic en el botón "Build Secure Application" o pulse **F5**. XLS Padlock compilará su libro y creará el archivo EXE de la aplicación.



Al finalizar, aparecerá un mensaje de confirmación:



Desde este cuadro de diálogo, puede ejecutar inmediatamente la aplicación, ver el registro de compilación (especialmente útil para solucionar errores) o abrir la carpeta de destino.

CONSEJO

XLS Padlock no modifica el archivo de su libro de Excel original durante la compilación. Conserve siempre una copia de seguridad de su libro original.

Solución de problemas de formato

Si su libro protegido muestra resultados extraños o problemas de formato, pruebe a activar la opción "Use Excel automation for formula protection" en la página "Formulas and Passwords".

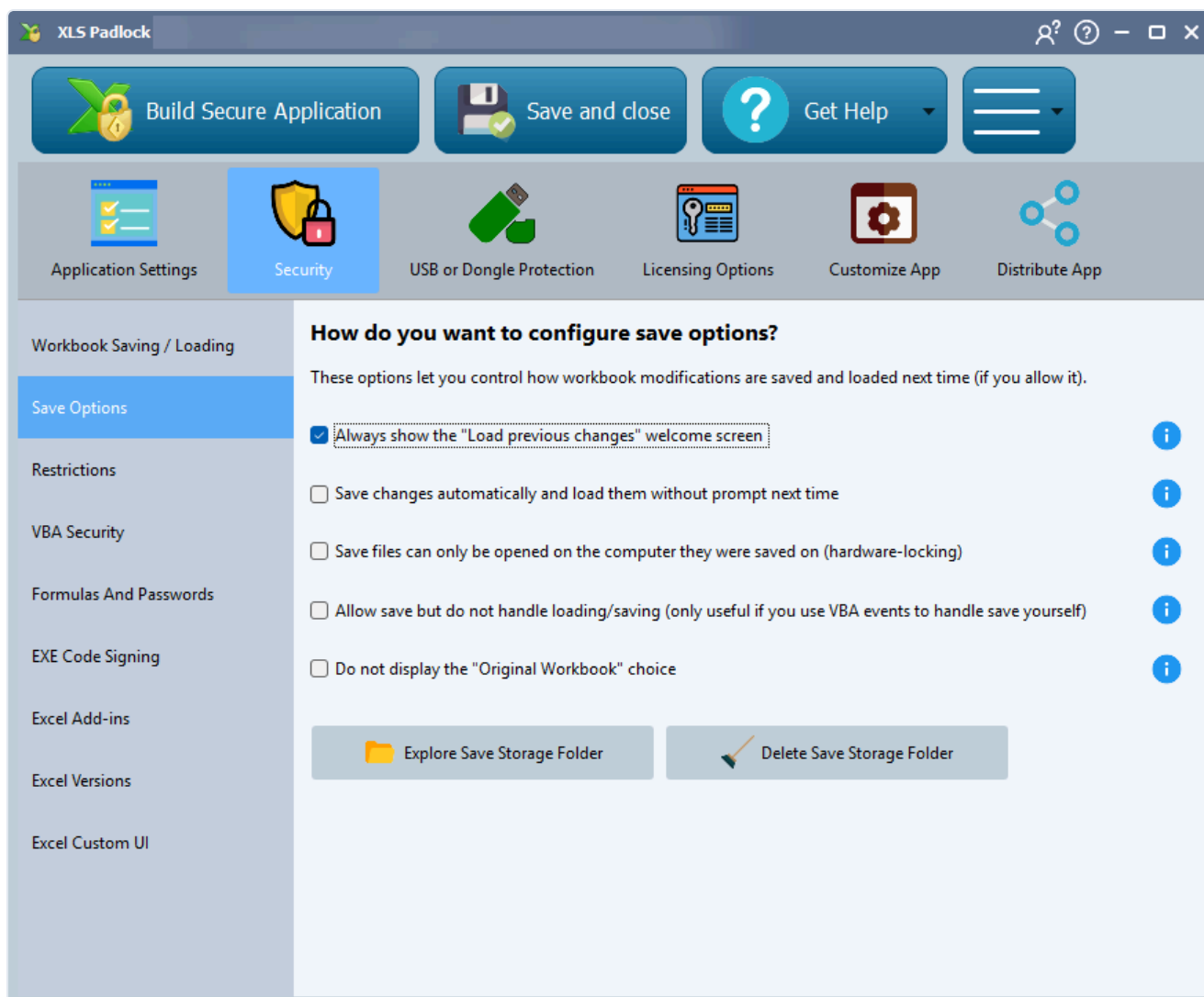
👉 Consulte también:

[Cómo distribuir un libro protegido](#)

Seguridad

XLS Padlock ofrece muchas formas de proteger su libro de Excel, incluidas opciones de guardado, restricciones de funciones y condiciones para abrir su aplicación, como exigir una [activation key](#) o la [presencia de una mochila \(dongle\) o memoria USB](#).

Este tema proporciona un resumen de las opciones de seguridad disponibles en "**Security**" dentro de XLS Padlock.



Guardado / carga del libro

Configure si los usuarios finales pueden guardar los cambios en el libro protegido. Cuando el guardado está habilitado, XLS Padlock crea archivos seguros `.XLSC` o `.XLSCE` que solo pueden abrirse con su aplicación, lo que protege los datos del usuario.

[➔ Más información sobre el guardado y la carga del libro](#)

Opciones de guardado

Ajuste con precisión el comportamiento de guardado con opciones como guardar automáticamente los cambios al cerrar, vincular por hardware los archivos guardados a un equipo concreto o mostrar siempre la pantalla de bienvenida para que los usuarios elijan entre empezar de cero y cargar un trabajo anterior.

→ [Explore las opciones avanzadas de guardado](#)

Restricciones

Restrinja funciones clave de Excel para reforzar la seguridad. Puede deshabilitar el menú contextual del botón derecho, la barra de fórmulas, los comandos de copiar y pegar, la impresión y mucho más para impedir que los usuarios accedan a partes sensibles de su libro o las modifiquen.

→ [Consulte las funciones de restricción](#)

Seguridad de VBA

Proteja su código VBA frente a su visualización o modificación. Puede bloquear el proyecto VBA con una contraseña que queda oculta para el usuario o bloquear por completo el acceso al editor de VBA (VBE), lo que inutiliza las herramientas para descifrar contraseñas. Para una protección máxima, utilice el VBA Compiler.

→ [Consulte los distintos medios de protección de VBA](#)

Fórmulas y contraseñas

Proteja sus fórmulas y la estructura de las hojas de cálculo. Utilice la protección de fórmulas de XLS Padlock para que los usuarios puedan interactuar con las fórmulas sin poder verlas ni copiarlas. También puede gestionar las contraseñas del libro y de las hojas de cálculo directamente desde la interfaz.

→ [Más información sobre la protección de fórmulas](#)

Firma de código del EXE

Firme digitalmente su archivo `.EXE` compilado para generar confianza entre sus usuarios y evitar las advertencias de Windows SmartScreen y del software antivirus. XLS Padlock puede automatizar el proceso de firma si dispone de un certificado de firma de código.

→ [Cómo firmar digitalmente el código de su aplicación](#)

Complementos de Excel

De forma predeterminada, XLS Padlock deshabilita la mayoría de los complementos por seguridad. Esta sección le permite volver a habilitar de forma selectiva determinados complementos COM o complementos habituales de Excel (como Analysis ToolPak o Solver) que su libro necesita para funcionar correctamente.

→ [Cómo gestionar los complementos de Excel](#)

Versiones de Excel

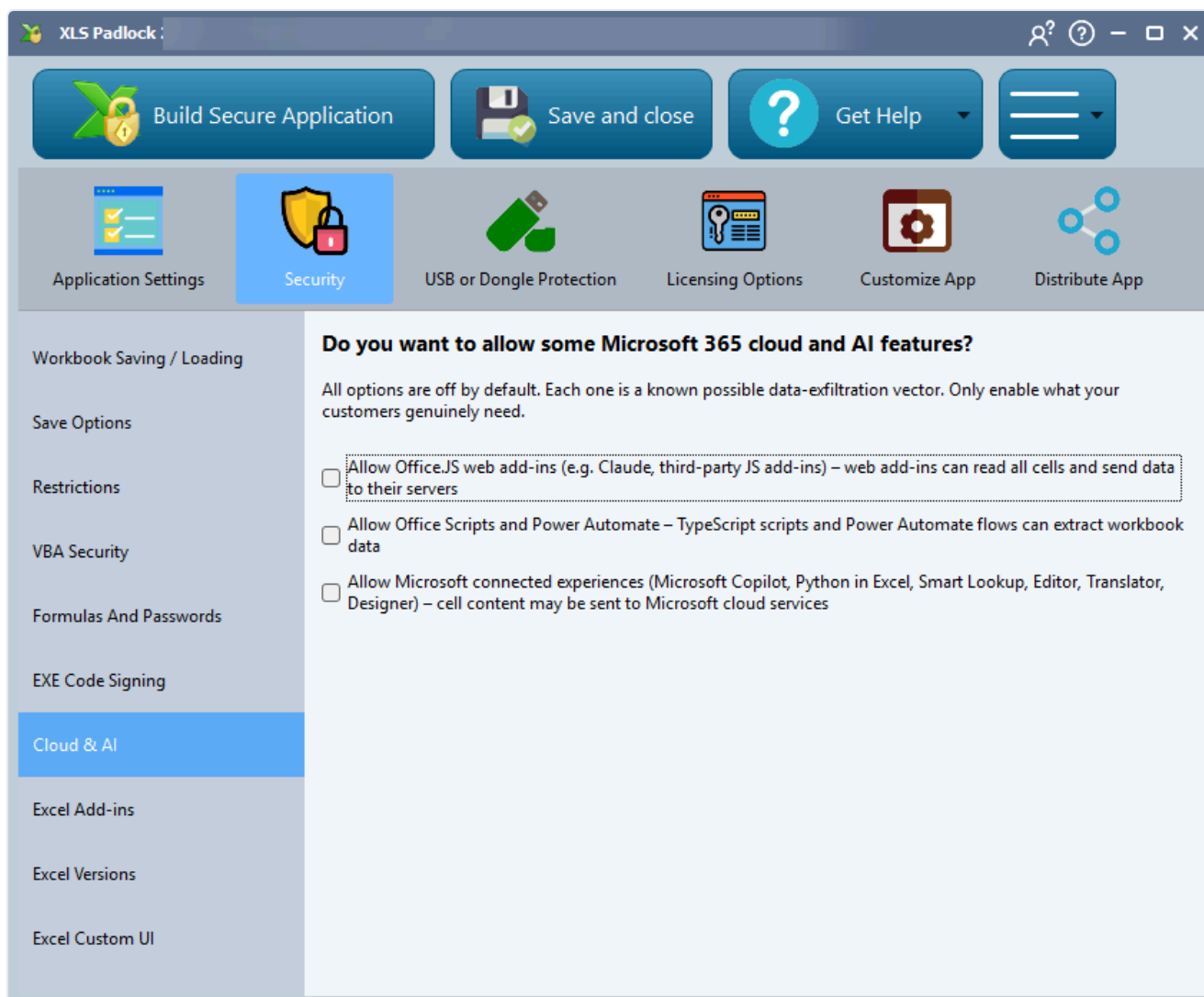
Especifique qué versiones de Microsoft Excel son compatibles con su aplicación protegida. Si un usuario intenta ejecutar el EXE en una versión no compatible, se mostrará un mensaje de error personalizable.

[→ Cómo exigir versiones concretas de Excel](#)

Opciones de nube e IA

Las versiones modernas de Excel incluyen varias funciones de nube e IA que pueden leer su libro y enviar su contenido a servidores externos. Para mantener su aplicación protegida sellada, XLS Padlock las bloquea todas de forma predeterminada. Usted decide, una por una, cuáles permitir.

Encontrará estos ajustes en el diseñador de XLS Padlock, en la pestaña **Security**, en la página **Cloud & AI**.



Por qué estas funciones se bloquean de forma predeterminada

Cada opción de esta página es una vía conocida para que los datos del libro salgan del equipo del usuario final. Un libro protegido está pensado para mantener sus fórmulas, datos y código VBA bajo su control, por lo que XLS Padlock parte de la posición más segura: todo desactivado. Active únicamente lo que sus clientes realmente necesiten.

Opciones disponibles

Las tres opciones están sin marcar (bloqueadas) de forma predeterminada.

Allow Office JS web add-ins

Los complementos web de Office JS (también llamados web add-ins) se ejecutan dentro de Excel y pueden leer todas las celdas del libro abierto y enviar esos datos a sus propios servidores. Esto incluye complementos de terceros como el plugin **Claude for Excel** y cualquier otro complemento JS instalado desde la tienda de Office o cargado de forma lateral (sideloaded).

Cuando esta opción está desactivada, los complementos web no pueden cargarse contra su libro protegido. Actívela solo si su aplicación depende de un complemento web concreto en el que confía.

Allow Office Scripts and Power Automate

Los Office Scripts (scripts de automatización en TypeScript) y los flujos de Power Automate pueden leer el contenido del libro y sacarlo de la aplicación. Deje esta opción desactivada a menos que su flujo de trabajo dependa de ellos.

Allow Microsoft connected experiences

Las experiencias conectadas de Microsoft (Microsoft connected experiences) envían el contenido de las celdas a los servicios en la nube de Microsoft. Esto abarca Microsoft Copilot, Python in Excel, Smart Lookup, Editor, Translator y Designer. Actívela solo si sus clientes necesitan estas funciones y acepta que el contenido de celdas correspondiente puede procesarse en la nube de Microsoft.

Notas

- Estas opciones se aplican a la aplicación compilada, no al diseñador de XLS Padlock.
- Bloquear estas superficies también oculta sus puntos de acceso en la cinta de Excel (por ejemplo, el menú emergente Add-ins, la pestaña Automate y el botón Copilot), de modo que no se invita a los usuarios finales a usar una función que está desactivada.
- Los complementos COM y los complementos integrados de Excel se gestionan por separado. Consulte [Cómo gestionar los complementos de Excel](#).

Mejorar la protección

Por qué compilar a un EXE no es suficiente

Cuando compila un libro en una aplicación protegida, el archivo de libro original (XLSX, XLSM, etc.) se cifra y se incrusta dentro del archivo .EXE resultante. Esto impide que los usuarios accedan directamente al archivo original.

La protección básica no es suficiente

Sin embargo, limitarse a compilar su libro proporciona solo una capa básica de seguridad. Piénselo como cerrar una puerta sin echar la llave. Para proteger de verdad su propiedad intelectual, debe habilitar las funciones de protección avanzada que ofrece XLS Padlock.

Para funcionar, XLS Padlock debe cargar su libro en la memoria de Excel. **Sin activar las capas de seguridad adicionales de XLS Padlock**, un atacante sofisticado podría llegar a extraer el libro original de la memoria mientras la aplicación se está ejecutando. Aunque esto no es una tarea trivial, pone de relieve la necesidad de una protección más sólida.

Haga que un libro extraído resulte inútil

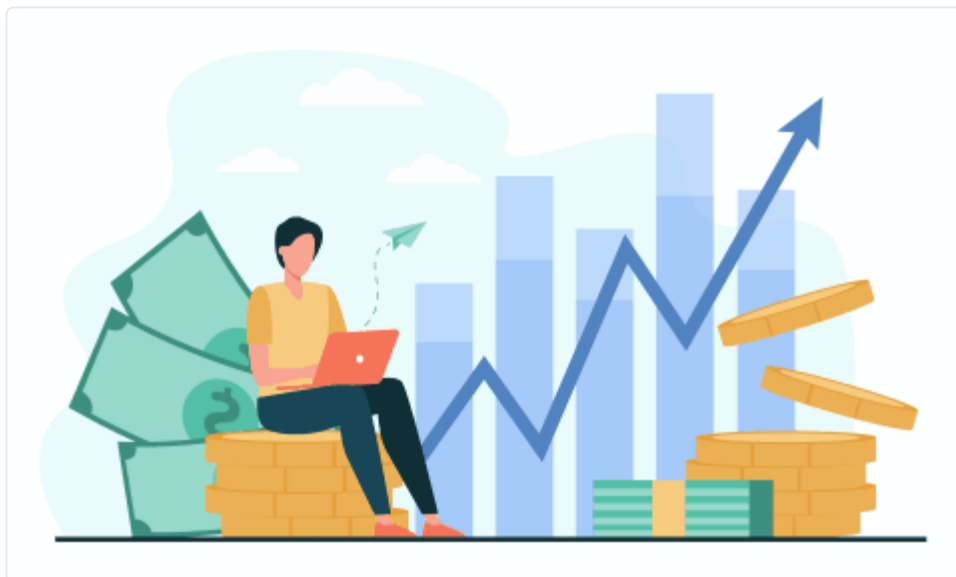
Cuando aplica correctamente las funciones de seguridad de XLS Padlock, el libro extraído resulta inútil. Aunque alguien extraiga el archivo, no funcionará correctamente. La protección es especialmente sólida en los libros que contienen fórmulas o código VBA, ya que estos componentes críticos están cifrados o compilados, lo que impide de forma eficaz su uso no autorizado o su manipulación.

Las funciones de seguridad más eficaces

Recomendamos habilitar las siguientes funciones para obtener la protección más sólida:

- [Proteja sus fórmulas con la protección de fórmulas](#)
- [Proteja su VBA con el VBA Compiler](#)
- [Evite los ataques habituales contra VBA y OLE](#)
- [Prohíba el acceso al editor de VBA \(VBE\)](#)
- [Proteja su libro con contraseña](#)
- [Deshabilite los complementos de Excel innecesarios](#)

Control de acceso al libro



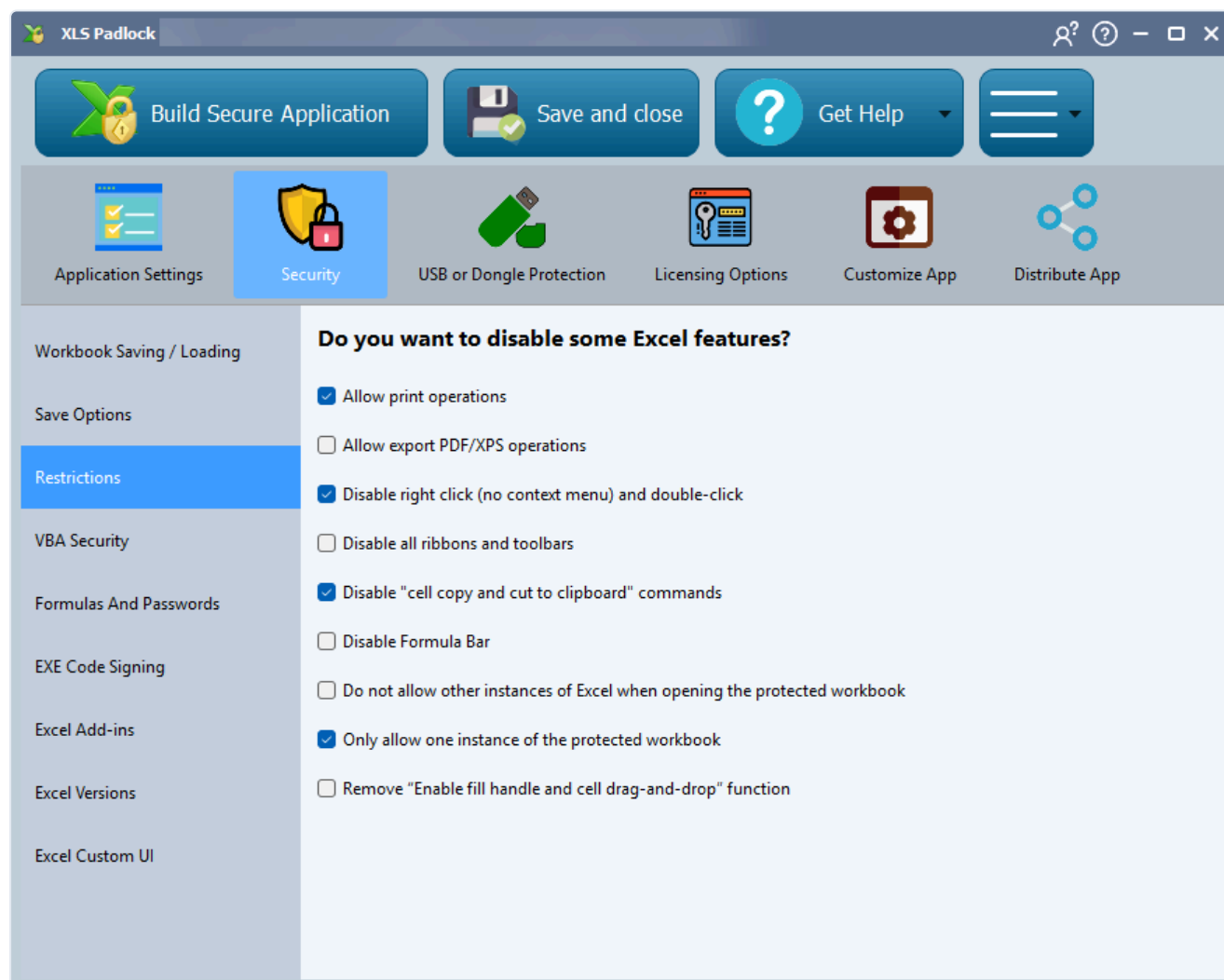
XLS Padlock ofrece numerosas funciones de seguridad para **restringir el acceso a sus libros de Excel únicamente a usuarios autorizados**. Puede generar claves de activación, configurar versiones de prueba con límite de tiempo y mucho más. Un objetivo principal de XLS Padlock es permitirle **vender licencias para sus libros**, ofreciendo funcionalidades para reducir la piratería y el uso compartido no autorizado.

Este tema ofrece un resumen de las funciones de seguridad dedicadas al control de acceso al libro:

- **Controle quién puede usar su libro:** [Configure claves de activación](#).
- **Evite el uso compartido vinculando una licencia a un único equipo:** Use [claves de activación bloqueadas por hardware](#).
- **Gestione las activaciones de forma remota y desactive el acceso:** Use [activación en línea](#), [desactivación](#) y [validación](#).
- **Cree versiones de prueba de su libro:** Aprenda [cómo crear libros de prueba](#).
- **Restrinja las acciones del usuario:** [Prohíba imprimir](#), [exportar](#), [hacer clic con el botón derecho](#) y [mucho más](#).
- **Exija un dispositivo físico para el acceso:** Vincule la aplicación a un [dongle](#) o [memoria USB](#).
- **Venda licencias y suscripciones en línea:** Intégrese con plataformas de comercio electrónico como [WooCommerce](#) o [FastSpring](#).

Restricciones del libro de Excel

Este tema cubre las opciones disponibles en la sección 'Security > Restrictions' de XLS Padlock, que le permiten restringir ciertas acciones del usuario en su libro protegido.



Allow Print Operations

Para evitar que los usuarios impriman su libro, desactive la opción **Allow print operations** en la configuración de [Security](#).

Si esta opción está desactivada, se bloqueará cualquier intento de impresión y el usuario verá el mensaje de error "Printing is not allowed".

Allow Export to PDF/XPS

Para evitar que los usuarios exporten el contenido del libro a archivos PDF o XPS, desactive la opción "Allow export PDF/XPS operations" en la configuración de [Security](#).

Disable Right-Click (and optionally Double-Click)

Para deshabilitar el menú contextual del botón derecho en las celdas de su libro, active la opción "Disable right-click" en la configuración de [Security](#).

Esto se usa habitualmente para evitar que los usuarios copien, peguen o modifiquen los datos y el formato de las celdas.

Disable All Ribbons and Toolbars

La opción **Disable all ribbons and toolbars** oculta todas las pestañas de la cinta de opciones integradas y la mayoría de los comandos del menú de Microsoft Office, creando una interfaz simplificada, similar a un quiosco, para su aplicación.

Esta función funciona con **Excel 2007 y posterior**. Puede aplicarla estableciendo la [versión mínima requerida de Excel](#) para su libro.

Disable "Cell Copy and Cut to Clipboard" Commands

Active esta opción para deshabilitar los comandos **Copy** y **Cut** para las celdas. Esto evita que los usuarios finales copien con facilidad datos de su libro al portapapeles para reutilizarlos en otras aplicaciones.

Disable Formula Bar

La opción **Disable Formula Bar** oculta la barra de fórmulas de Excel. Además, se deshabilita la opción correspondiente en el menú Ver de Excel, de modo que los usuarios finales no puedan volver a activarla.

Para mayor seguridad, también recomendamos que [deshabilite el acceso al editor de VBA](#).

Do Not Allow Other Instances of Excel

Cuando está activada, esta opción comprueba si ya hay otra instancia de Excel en ejecución cuando un usuario inicia su libro compilado. Si se encuentra un proceso de Excel existente, un mensaje pedirá al usuario que lo cierre antes de continuar.

Esto garantiza que la instancia de Excel que ejecuta su libro protegido sea la primera y la única, proporcionando un entorno más controlado.

Esta opción se encuentra en la configuración de [Security](#).

Only Allow One Instance of the Protected Workbook

La opción "Only allow one instance of the protected workbook", que se encuentra en la configuración de [Security](#), evita que los usuarios finales ejecuten más de una instancia de su aplicación protegida al mismo tiempo. Si un usuario intenta abrir la aplicación de nuevo mientras ya está en ejecución, la ventana existente se traerá al primer plano. Esta función impide eficazmente que los usuarios abran varias instancias del mismo libro protegido.

Nota de compatibilidad

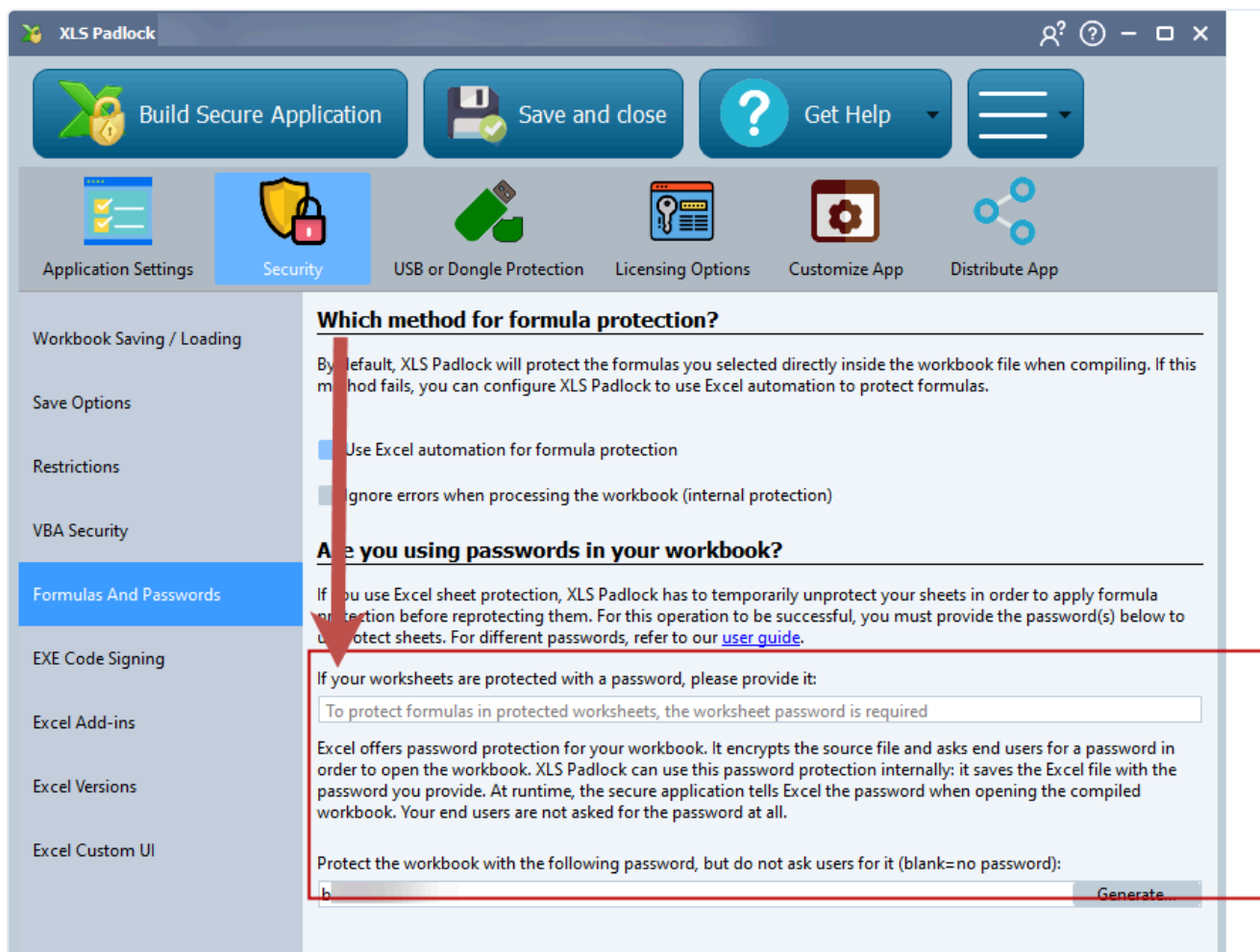
En versiones de Excel anteriores a 2013, se mostrará un cuadro de mensaje con el texto "An instance of this application is already running" en lugar de enfocar la ventana existente.

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

Esta opción "Remove "Enable fill handle and cell drag-and-drop" function" impide que los usuarios finales usen el controlador de relleno y el arrastrar y soltar de celdas en el libro protegido. Es lo mismo que en las opciones generales de Excel ("Review", "Check accessibility", "Options accessibility", "Advanced", "Enable fill handle and cell drag-and-drop"), pero le permite anular la elección del usuario local.

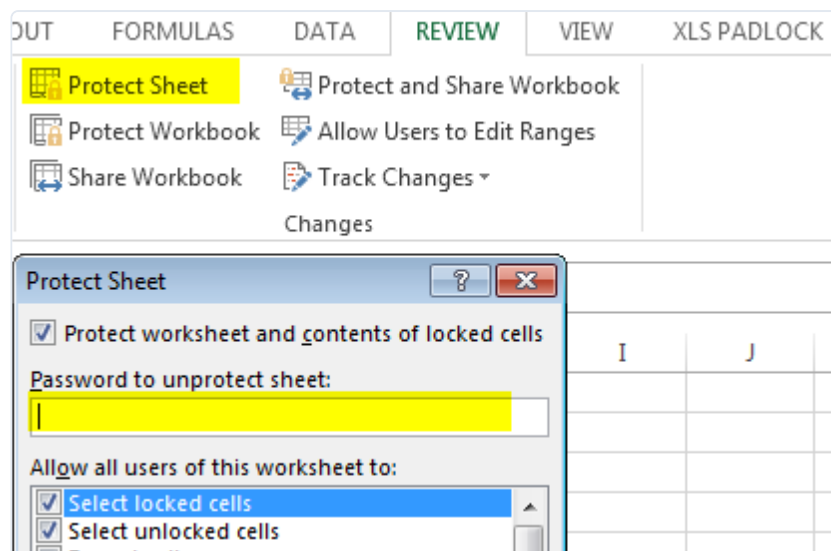
Protección por contraseña de las hojas

XLS Padlock le permite gestionar las contraseñas tanto del libro como de las hojas de cálculo para mejorar la seguridad y la funcionalidad de su aplicación protegida. Esta guía explica cómo configurar estos ajustes de contraseña.



Protección con contraseña de la hoja de cálculo

Si usa la protección de hojas integrada de Excel para restringir el acceso a las celdas, XLS Padlock necesita conocer su contraseña para gestionar sus propias capas de seguridad.



Cuando la [protección de fórmulas](#) de XLS Padlock está activa, debe desproteger temporalmente sus hojas, aplicar su protección y luego volver a protegerlas. Para realizar esta operación correctamente, debe proporcionar la contraseña usada para la protección de hojas en Excel.

NOTA

XLS Padlock vuelve a aplicar automáticamente las mismas opciones de protección de hojas que definió en el libro original.

Si usa una única contraseña para todas las hojas de cálculo protegidas, introdúzcala en el campo "If your worksheets are protected with a password, please provide it".

Si usa contraseñas diferentes para hojas diferentes, debe proporcionarlas usando el siguiente formato JSON:

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

Por ejemplo:

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

Protección silenciosa con contraseña del libro

Excel ofrece una función para cifrar todo el libro con una contraseña. XLS Padlock puede gestionar esta contraseña por usted, proporcionando una capa adicional de seguridad sin pedir nada a sus usuarios finales.

Cuando establece una contraseña en el campo "Protect the workbook with the following password", XLS Padlock guarda el libro con esa contraseña. En tiempo de ejecución, la aplicación segura proporciona automáticamente la contraseña a Excel al abrir el libro. **A sus usuarios finales nunca se les pide la contraseña.**

Esta combinación aumenta la seguridad de su aplicación al superponer la protección de XLS Padlock con el cifrado nativo de Excel.

Puede usar el botón **Generate** para crear una contraseña fuerte y aleatoria.

No cambie la contraseña después de la distribución

Una vez que haya distribuido una aplicación protegida con una contraseña de libro específica, no la cambie. Si lo hace, los usuarios finales no podrán abrir sus datos guardados previamente, porque la aplicación no podrá descifrarlos.

Si deja el campo de contraseña en blanco, esta función se deshabilitará.

Combinar la protección de Excel y XLS Padlock

La protección de fórmulas de XLS Padlock es totalmente compatible con las funciones de protección de hojas integradas en Excel, como las celdas bloqueadas u ocultas. Cuando utiliza ambas, las celdas protegidas no mostrarán nada en la barra de fórmulas, y las llamadas a la función `PLEvalForm` subyacente permanecerán ocultas.

	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	$[\ln(S/X) + (r + 0,5 \delta^2) \times t] / \sigma \sqrt{t}$	2.52

Advertencia sobre dependencias

Las fórmulas protegidas que dependen de otras celdas fallarán si esas celdas se ocultan mediante el atributo de formato de celda "Hidden" de Excel. Evite usar el atributo oculto en celdas que sean dependencias de sus fórmulas protegidas.

Véase también:

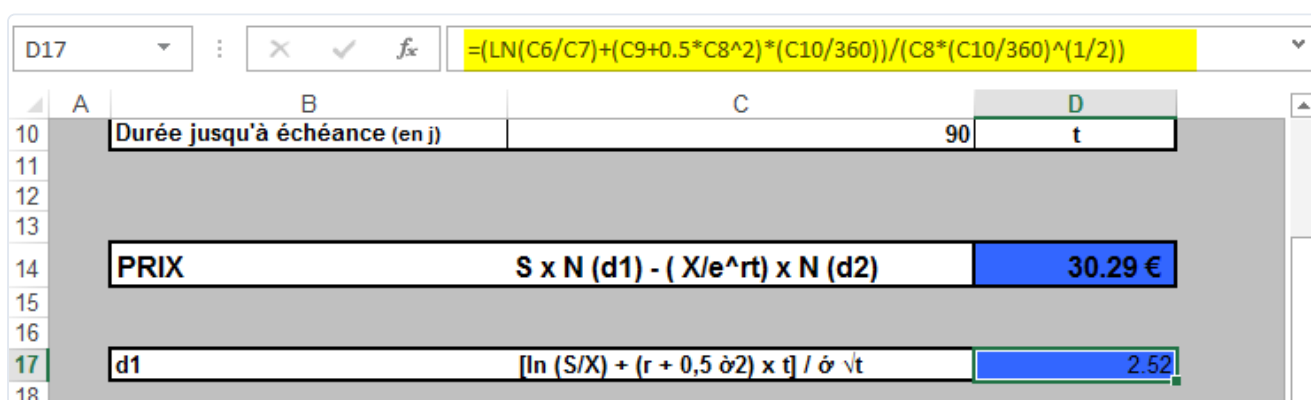
- [Desactivar la protección de fórmulas](#)

Protección de fórmulas: Excel frente a XLS Padlock

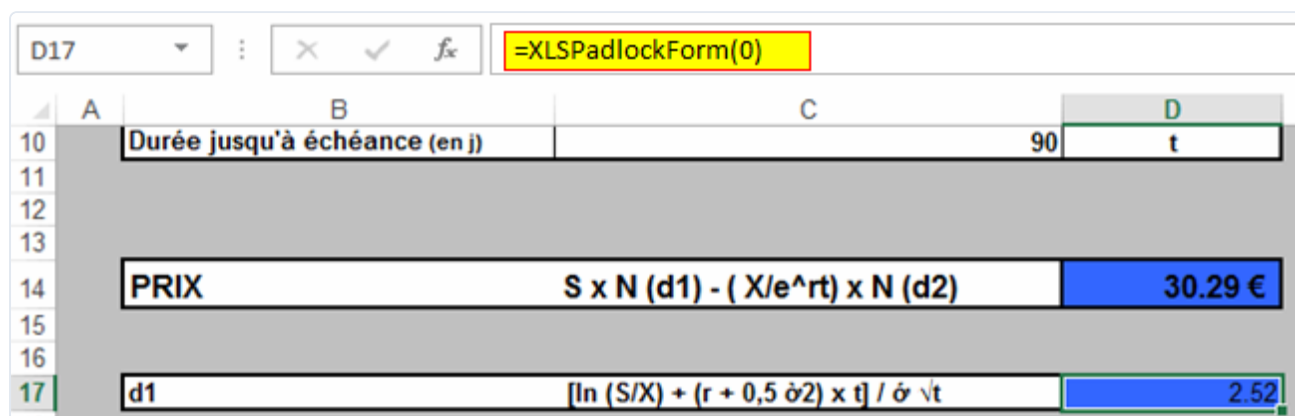
Aunque Excel permite ocultar fórmulas en una hoja protegida, esta protección es limitada. XLS Padlock va mucho más allá al reemplazar sus fórmulas con una función segura que solo está disponible cuando su aplicación está en ejecución.

En el libro compilado, las fórmulas se reemplazan en la barra de fórmulas por funciones anónimas como `=PLEvalFormD(0;0)` o `=PLEvalFormD(1;COUNT($F7))`. Sin embargo, los cálculos seguirán funcionando como se espera.

Antes de la protección, cualquiera puede ver su fórmula:

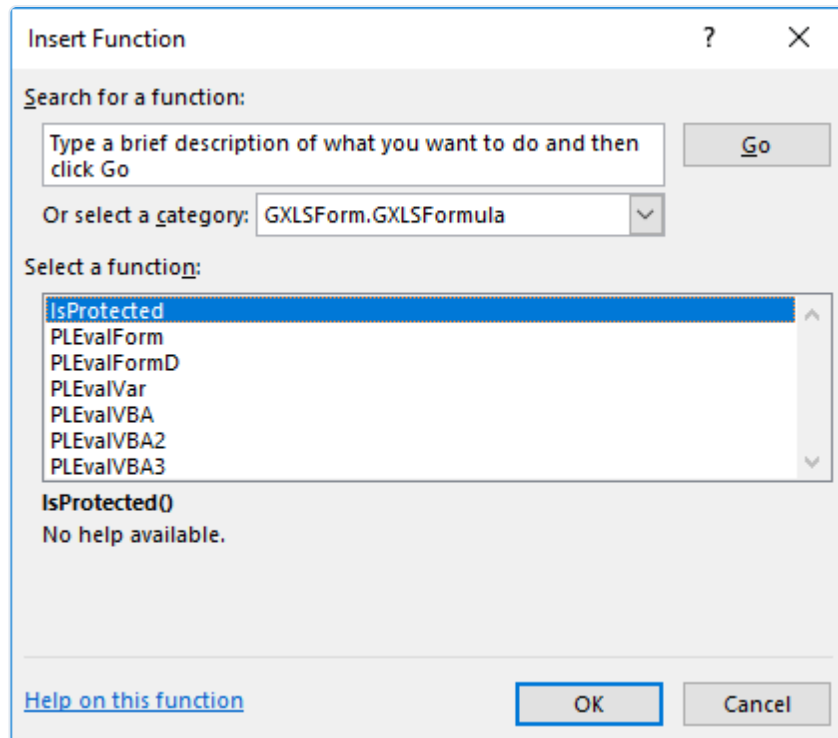


Después de la protección con XLS Padlock, la fórmula se reemplaza automáticamente sin dejar de ser funcional:



Esto aumenta considerablemente la seguridad de su libro. Dado que las fórmulas originales se eliminan del archivo del libro, las celdas protegidas solo pueden funcionar correctamente cuando el libro se abre dentro de la aplicación segura creada por XLS Padlock.

Incluso en tiempo de diseño, puede ver que las funciones de seguridad de XLS Padlock como `PLEvalForm` y `PLEvalFormD` son reconocidas por Excel:



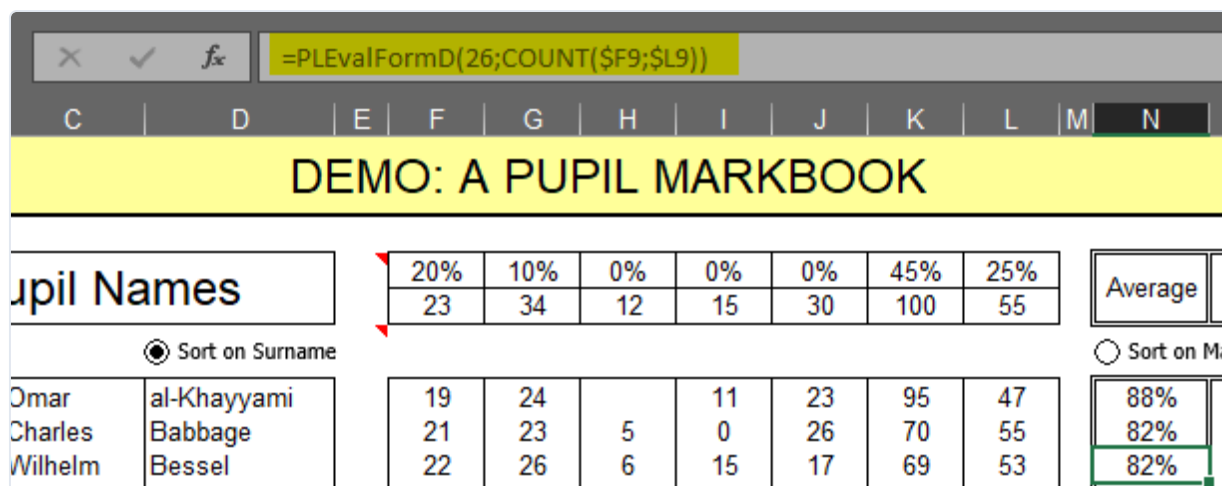
👉 Obtenga más información sobre [cómo proteger celdas con XLS Padlock](#).

Proteger fórmulas con XLS Padlock

Además de la protección de celdas de Excel, **recomendamos encarecidamente proteger las fórmulas importantes con la propia protección de fórmulas de XLS Padlock.**

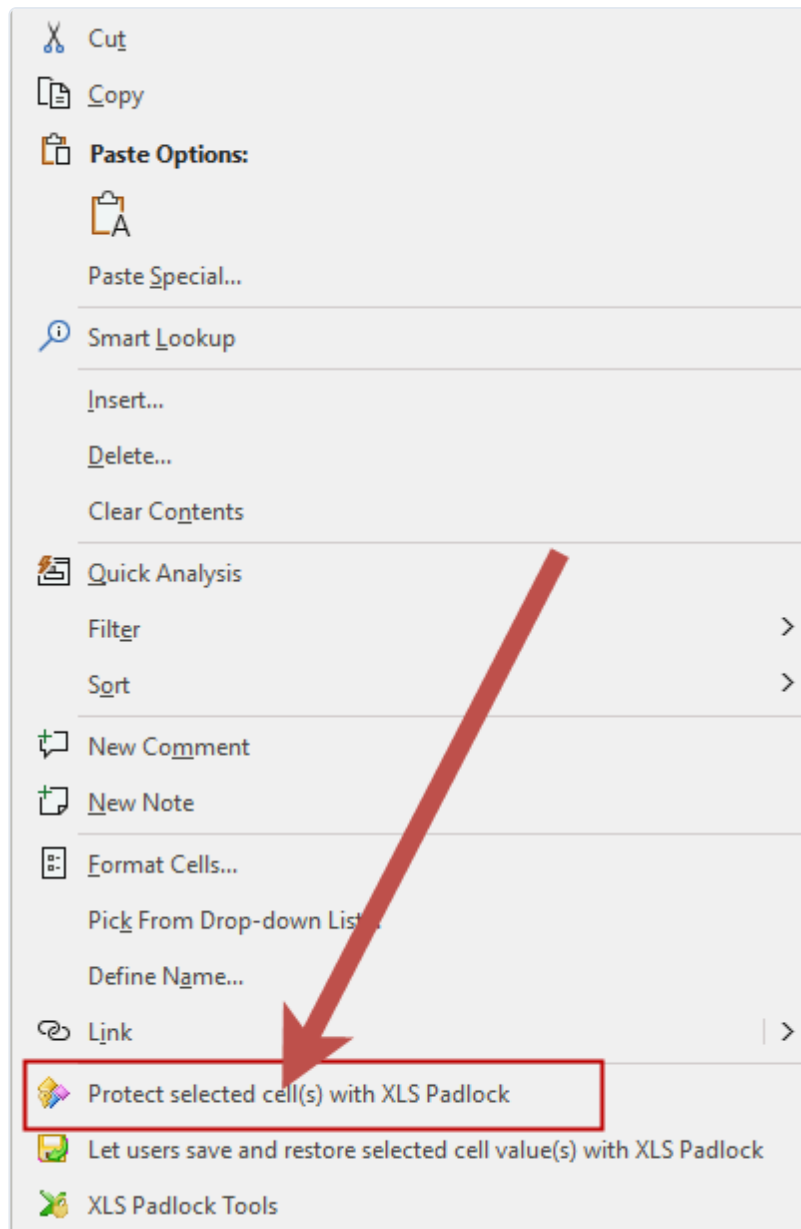
Tras la protección con XLS Padlock, sus fórmulas ya no aparecerán en la barra de fórmulas, **sin dejar de ser plenamente funcionales.** En su lugar, se muestra una llamada a función críptica como

PLEvalFormD :



👉 El objetivo es dejar el libro inservible si alguien consigue [extraer el archivo del libro del EXE compilado](#). Dado que las **fórmulas protegidas por XLS Padlock ya no están en el archivo del libro**, el libro extraído no funcionará correctamente.

XLS Padlock le permite decidir qué celdas proteger, y puede **seleccionar y proteger varias celdas a la vez.**



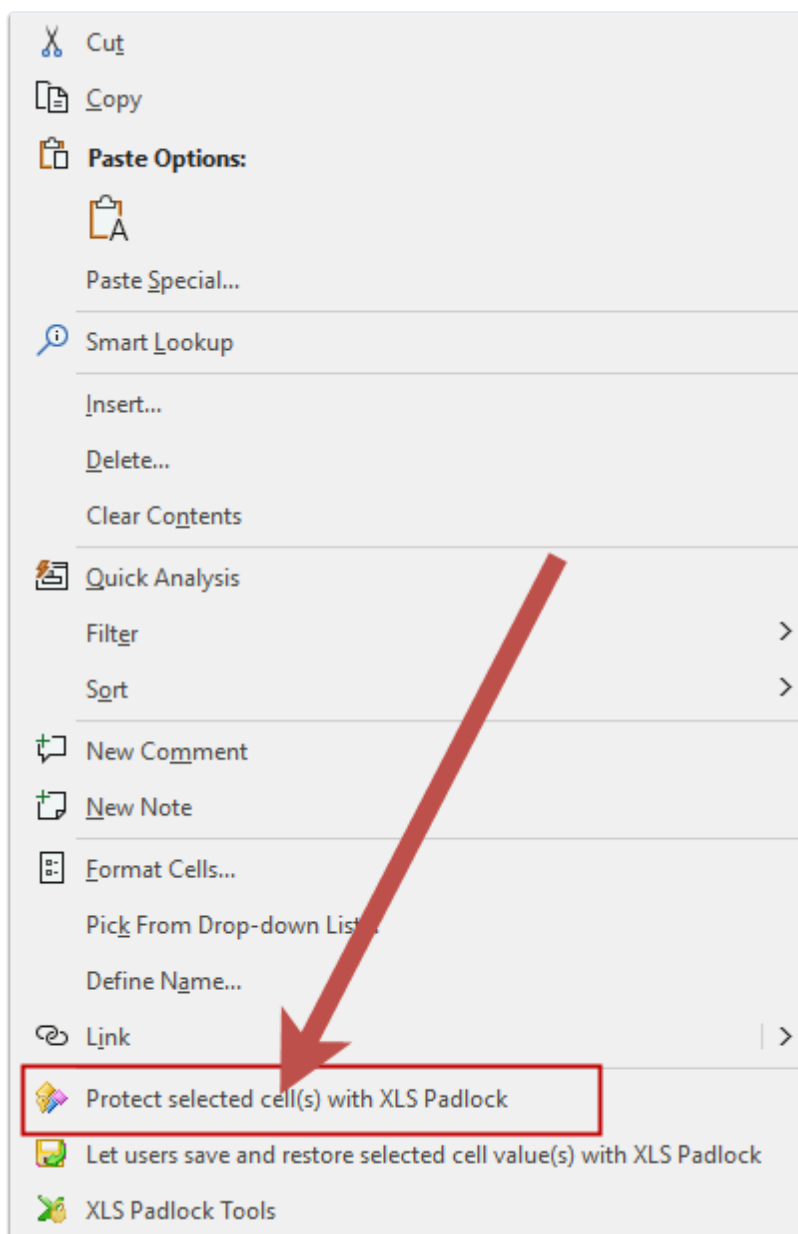
👉 Consulte [cómo seleccionar las celdas que se van a proteger](#)

Véase también:

- [Obtenga más información sobre cómo proteger sus fórmulas de forma eficiente](#)
- [Desactivar los complementos comunes de Excel](#)
- [Usar el compilador de VBA para una protección aún más sólida](#)

Marcar celdas para proteger

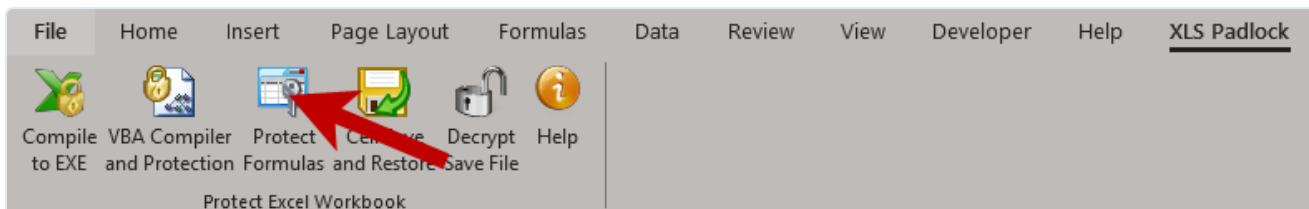
Para proteger una fórmula, simplemente **haga clic con el botón derecho en una o varias celdas** que contengan las fórmulas que desea ocultar y seleccione **“Protect selected cell(s) with XLS Padlock”** en el menú contextual.



A continuación, XLS Padlock confirmará que las celdas están marcadas para su protección.

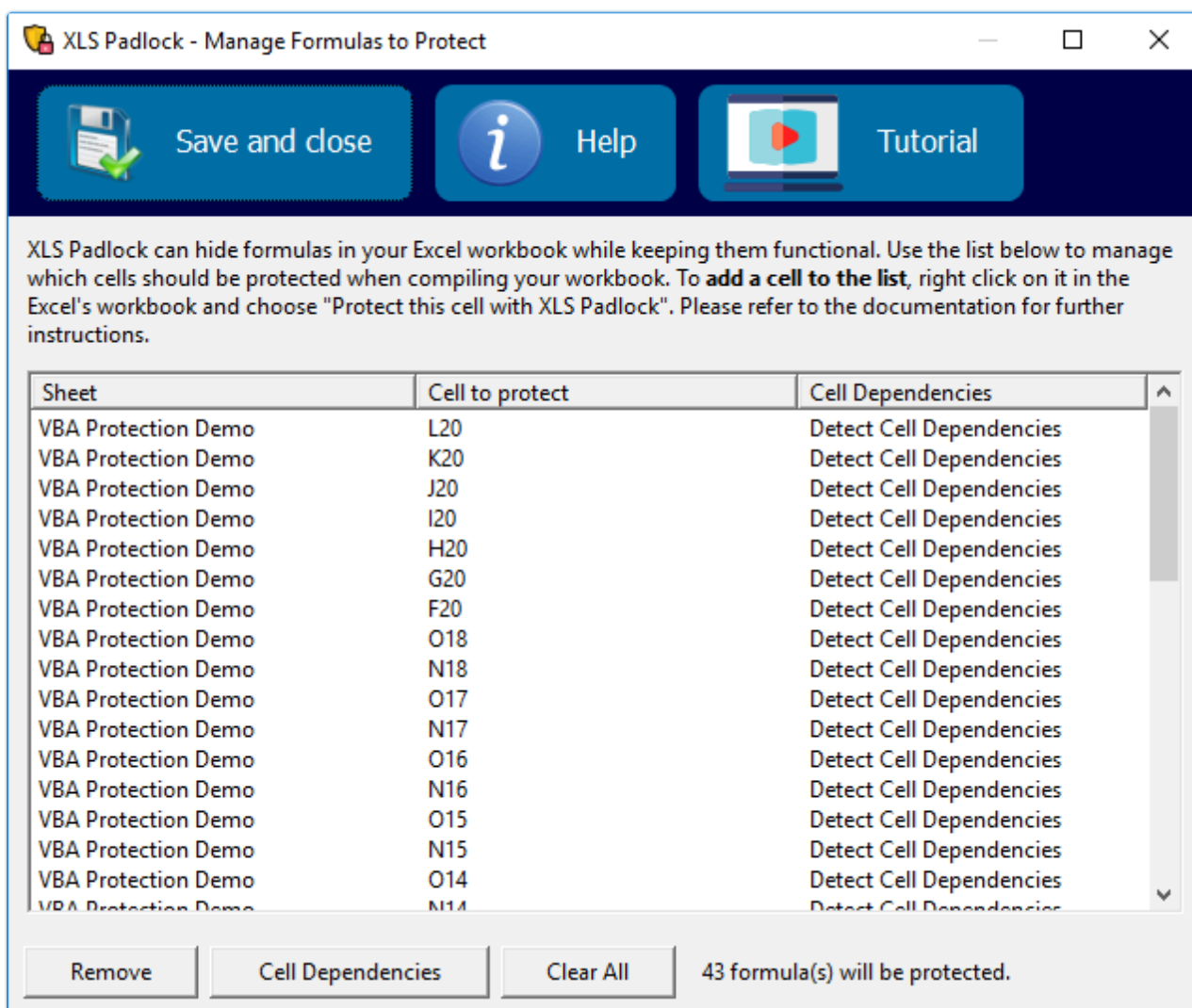
Resumen de las celdas protegidas

Para ver un resumen de todas las celdas protegidas, haga clic en **“Protect Formulas”** en la pestaña o el menú de XLS Padlock:



Esto abre una lista de todas las celdas configuradas para su protección. En esta ventana, puede ajustar el comportamiento de la protección, quitar la protección de celdas concretas o vaciar la lista completa.

Cuando compila su libro, XLS Padlock reemplaza todas las fórmulas de la lista por llamadas a funciones genéricas como `PLEvalForm(N)` y `PLEvalFormD(N, ...)`. **Sus celdas siguen siendo funcionales, pero los usuarios finales no pueden descubrir las fórmulas subyacentes.** Las fórmulas originales ya no existen en el libro compilado; las gestiona el propio EXE.



Dependencias de celdas

El botón **Cell Dependencies** le permite controlar cómo se aplica la protección. Hay dos opciones disponibles: "Detect Cell Dependencies" y "No".

De forma predeterminada, XLS Padlock detectará todas las referencias de celdas y los nombres de rango de sus fórmulas (dependencias de celdas) y generará una función anónima que contiene estas referencias. Esto permite que Excel recalculé correctamente las celdas protegidas. Por ejemplo, si la fórmula que se va a proteger es `=A3^2`, XLS Padlock generará una función como: `PLEvalFormD(1, COUNT(A3))`.

Si XLS Padlock no consigue proteger una celda, puede elegir “**No**” en su opción “Cell Dependencies”. En este caso, se utilizará una sencilla función genérica `PLEvalForm(N)`.

Limitaciones de la protección

Algunas fórmulas complejas no son compatibles con la protección y fallarán, mostrando un `#ERROR!` o `#VALUE!` en la celda. Pruebe sus fórmulas a fondo antes de distribuir su libro protegido. Preste especial atención a las siguientes limitaciones:

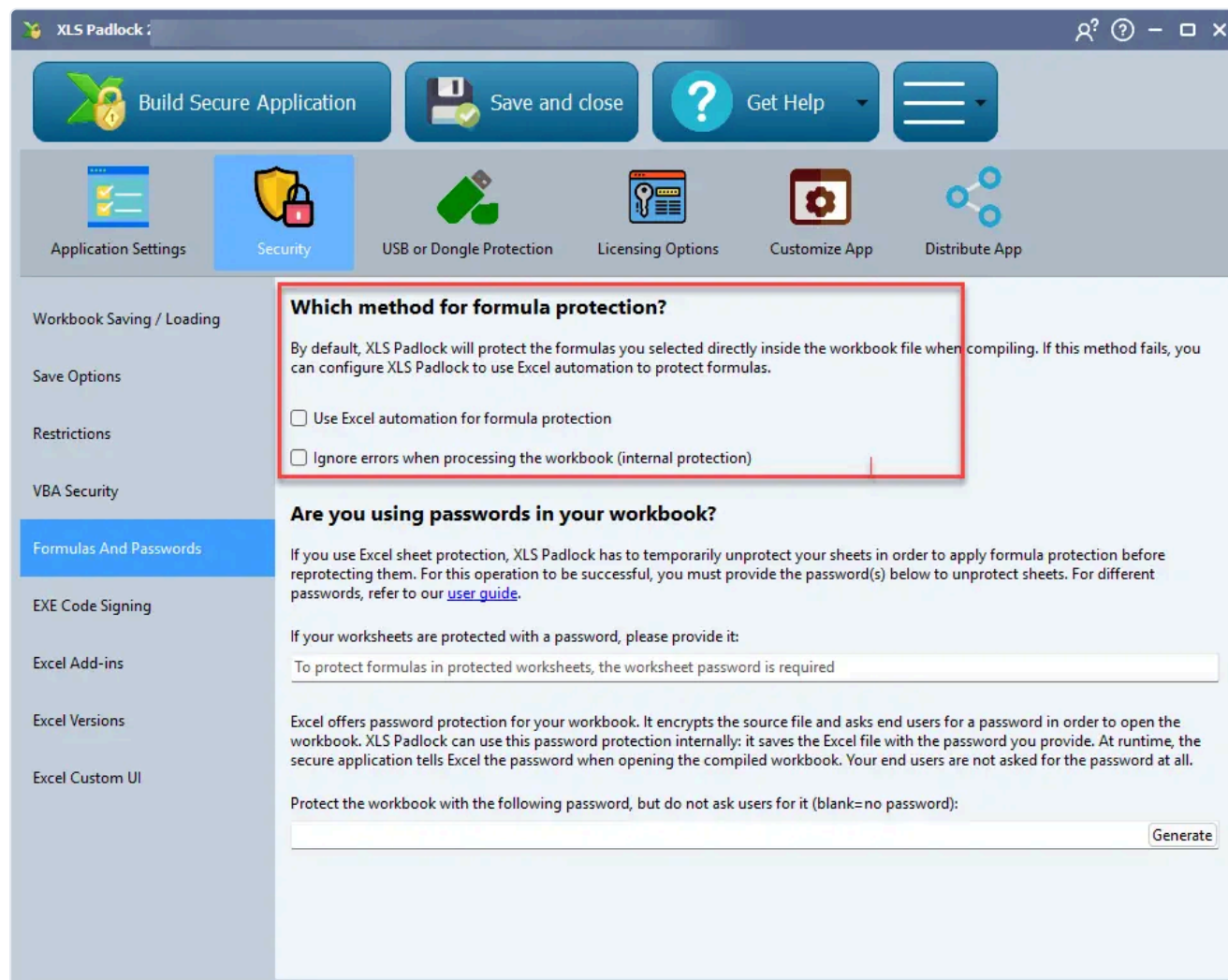
- La longitud de la fórmula debe ser inferior a 256 caracteres.
- La función `INDIRECT` no es compatible.
- Las fórmulas no deben contener ninguna función definida por el usuario de VBA, solo funciones normales de Excel.
- La protección de fórmulas debe usarse con moderación. Proteja únicamente las fórmulas más importantes de las que depende su libro. No se recomienda proteger varios miles de fórmulas, ya que aumentará el tamaño de su archivo EXE y puede ralentizar Excel en equipos antiguos.
- **La validación de datos de celdas y las reglas de formato condicional no son compatibles.**

👉 Vea también

- [Combinar la protección de hojas de Excel y la protección de XLS Padlock](#)

Método de protección de fórmulas

Cuando activa la protección de fórmulas, XLS Padlock puede usar uno de dos métodos:



1. **Direct Modification (Recommended)**: este método modifica las fórmulas directamente dentro del archivo de origen del libro. Es el método predeterminado y funciona de forma fiable en la mayoría de los casos.

2. **Excel Automation**: este método utiliza la automatización para controlar Excel y aplicar la protección.

Si el método recomendado falla, puede forzar a XLS Padlock a usar el método de automatización activando "**Use Excel automation for formula protection**". Este método también se utiliza automáticamente para libros binarios (XLSB) y archivos muy grandes.

Como último recurso, puede activar "[Ignorar errores al procesar el libro](#)", pero asegúrese de probar a fondo su aplicación compilada si activa esta opción.

Desactivar la protección de fórmulas

Dado que la protección de fórmulas de XLS Padlock es incompatible con Microsoft Excel 2000, debe activar la opción **Disable formula and VBA protection** en la página de Seguridad si necesita crear aplicaciones compatibles con esta versión.

El VBA Compiler también se desactiva

Al activar esta opción, también se desactivará el compilador de código VBA.

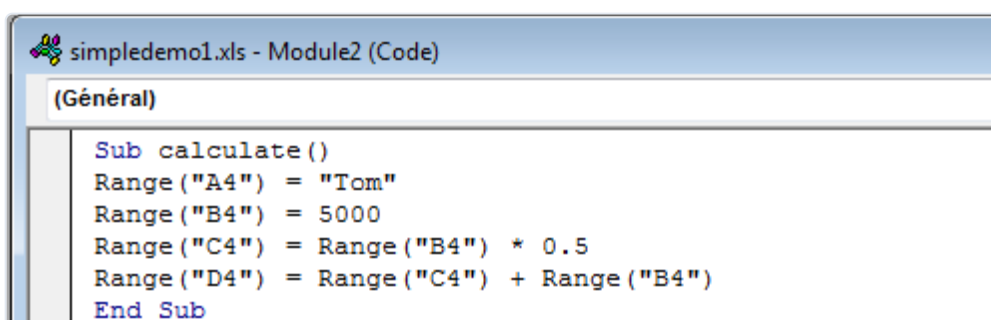
Acerca del VBA Compiler integrado

XLS Padlock incorpora un **compilador VBA integrado** que transforma sus scripts Basic en bytecode seguro, haciéndolos inaccesibles para los usuarios finales. Compilar partes de sus macros VBA elimina el código fuente original, haciendo imposible copiarlo.

Tutorial en vídeo

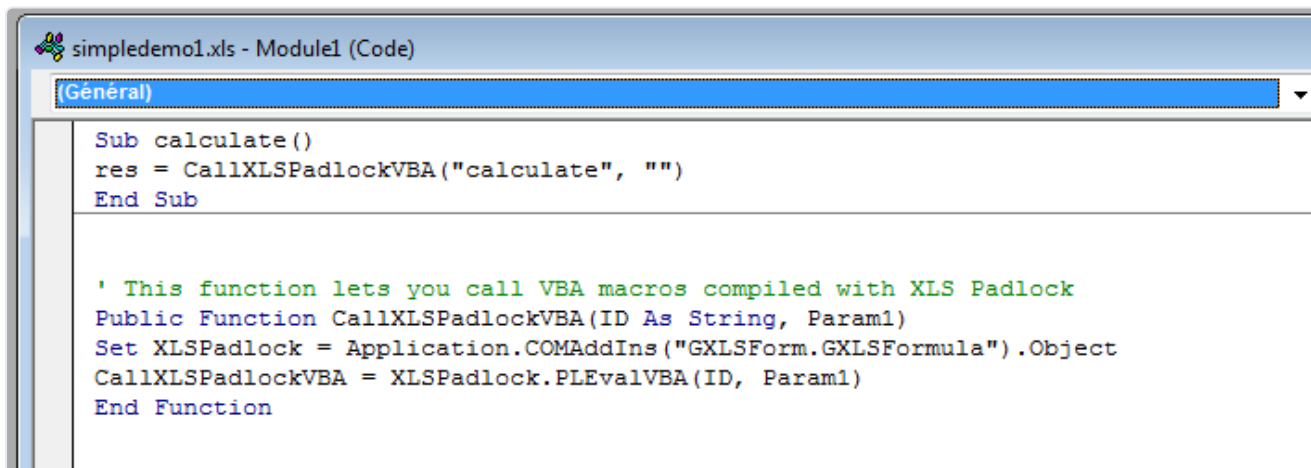
Para ver un tutorial sobre cómo compilar una función VBA, puede [ver este vídeo](#).

Por ejemplo, considere este código original:



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

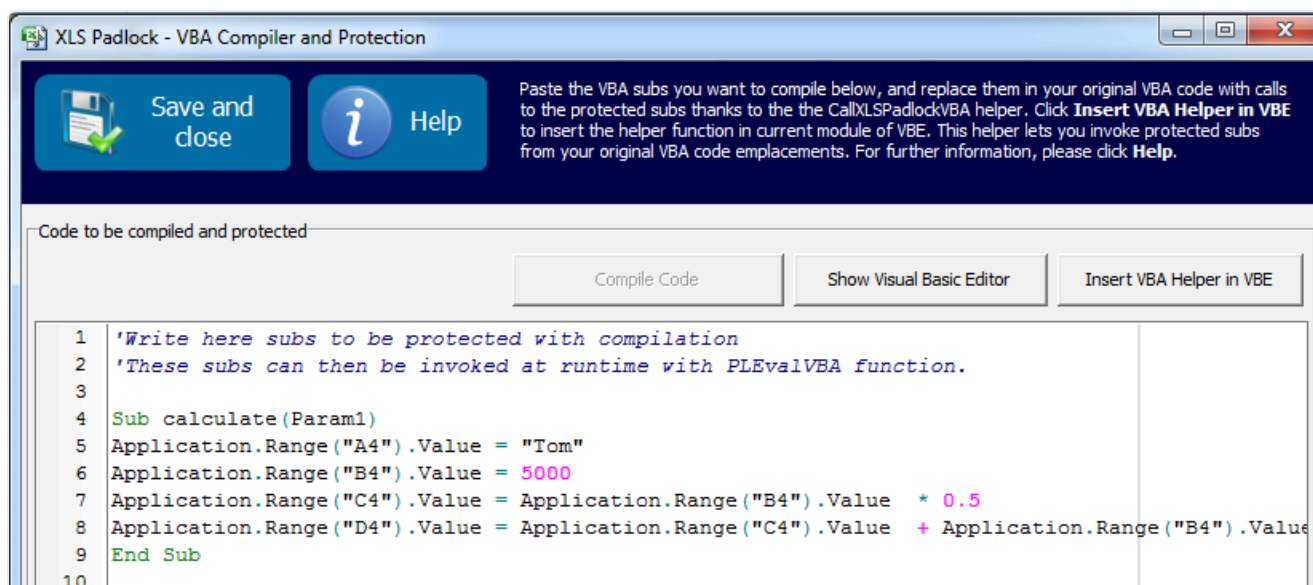
Una vez protegido, el sub `calculate()` original se sustituye por una llamada a una función interna de XLS Padlock que ejecuta el bytecode compilado:



```
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
```

El código original se traslada al editor VBA de XLS Padlock y se compila desde allí:



Tenga en cuenta que pueden ser necesarias algunas modificaciones, ya que el compilador requiere el uso explícito del objeto `Application` para [acceder a los objetos de Excel](#).

El compilador no es un simple ofuscador; transforma por completo su código VBA en binario y lo almacena de forma segura dentro de la aplicación. Cuando se combina con la [protección por contraseña de su proyecto VBA](#), esto deja inservibles las herramientas de descifrado de contraseñas, ya que no pueden acceder al archivo XLS original.

Limitaciones del compilador

- Es posible que deba modificar su código VBA original para que se compile, ya que el compilador VBA de XLS Padlock no es tan avanzado como el intérprete VB de Microsoft.
- Por lo general, las macros completas no se pueden compilar, solo partes de ellas.
- Algunos objetos y propiedades predeterminadas no son accesibles.

👉 Véase también: [\[Aprenda a escribir y compilar código VBA seguro\]\(#chapter-writing-and-compiling-secure-vba-code\)](#)

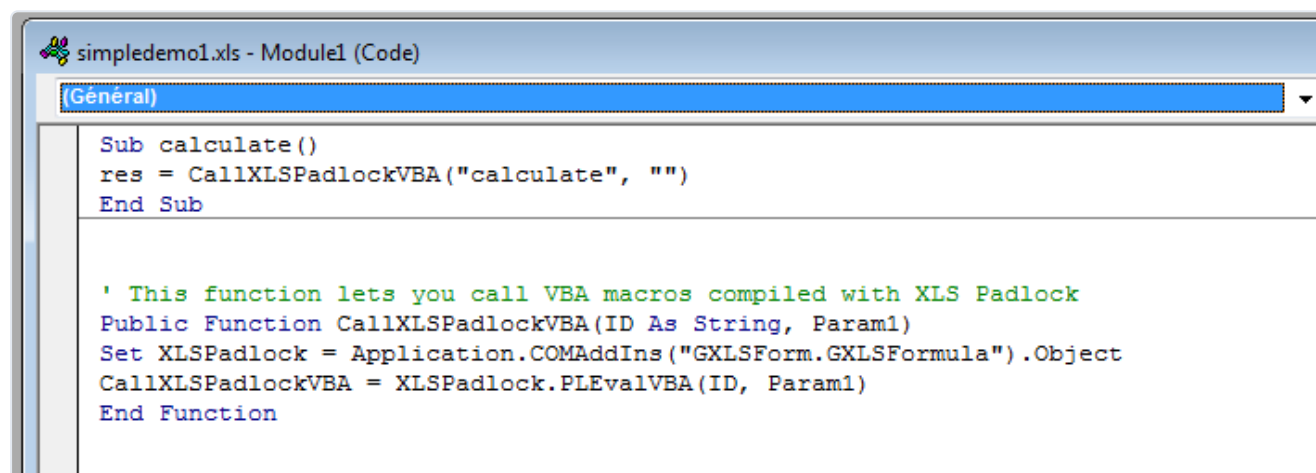
Protección real del código VBA

Aunque algunas herramientas ofrecen ofuscación de VBA, solo hacen que su código sea difícil de leer. No consiguen ocultar la lógica central ni impedir que se copie.

👉 XLS Padlock va más allá al permitirle **eliminar por completo el código VBA de su libro, sin dejar de mantenerlo funcional**. Esto es posible gracias al compilador VBA integrado.

El **compilador VBA de XLS Padlock convierte su código VBA en bytecode que solo se puede ejecutar dentro de la aplicación segura**. Si alguien logra [acceder al archivo del libro](#), no encontrará el código original porque sencillamente no está ahí. En su lugar, el **bytecode compilado se almacena de forma segura dentro del EXE**.

Por ejemplo, la macro `calculate()` que aparece a continuación contenía originalmente código VBA propietario. Tras la compilación, la macro del libro solo contiene una llamada al código compilado. La lógica original ha desaparecido.



```
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
```

Esto proporciona una **protección robusta** porque el código VBA original ya no existe en el libro. Ha sido sustituido por bytecode que no se puede copiar ni utilizar en otro libro de Excel.

La contrapartida es que debe **transferir manualmente al editor VBA de XLS Padlock el código VBA que desea proteger**. La mejor práctica consiste en identificar las partes más críticas de su código, aquellas que dejarían su libro no funcional si faltasen, y compilar solo esas secciones.

👉 Véase también: [Obtenga más información sobre cómo trabajar con nuestro compilador VBA](#)

Referencia de sintaxis

Este documento ofrece una referencia de la sintaxis VBA admitida por el [compilador VBA de XLS Padlock](#).

Estructura del script

La estructura del script se compone de declaraciones de funciones y subs.

```
SUB DoSomething
    CallSomething
END SUB

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

Las instrucciones en una sola línea se pueden separar con el carácter `:`.

Comentarios

Se pueden insertar comentarios dentro de un script. Puede usar el carácter `'` o `REM`. El comentario se extiende hasta el final de la línea.

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

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

Identificadores

Los nombres de identificadores (variables, funciones, procedimientos, etc.) deben empezar por un carácter (a-z, A-Z) o un guion bajo `_`. Pueden ir seguidos de caracteres alfanuméricos o guiones bajos. Los nombres no pueden contener ningún otro carácter ni espacios.

Variables

No es necesario declarar variables en un script, pero si lo desea, puede declarar una variable mediante la directiva `DIM`.

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

También puede declarar variables globales como `PRIVATE` o `PUBLIC` .

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

Las variables declaradas con `DIM` son públicas de forma predeterminada. Las variables privadas no son accesibles desde otros scripts. Las variables también se pueden inicializar al declararlas:

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

Instrucciones de asignación

Las instrucciones de asignación utilizan el operador `=` para asignar un valor o el resultado de una expresión a una variable o a una propiedad de objeto.

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

Matrices

El compilador ofrece compatibilidad básica con constructores de matrices y matrices de tipo variante. Para construir una matriz, use los caracteres `[` y `]` . Puede construir una matriz de varios índices anidando constructores de matrices.

Las matrices del compilador tienen índice de base 0.

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

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

Matrices dinámicas:

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

Índices

Las cadenas, las matrices y las propiedades de matriz se pueden indexar mediante los caracteres [y]. Por ejemplo, si Str es una variable de cadena, la expresión Str[3] devuelve el tercer carácter de la cadena.

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

Palabras clave y operadores

La sintaxis Basic admite:

- **Declaraciones:** SUB...END SUB , FUNCTION...END FUNCTION
- **Directivas:** BYREF , DIM
- **Condicionales:** IF...THEN...ELSE...ELSEIF...END IF , SELECT CASE...END SELECT
- **Bucles:** FOR...TO...STEP...NEXT , DO...WHILE...LOOP , DO...LOOP...WHILE , DO...UNTIL...LOOP , DO...LOOP...UNTIL
- **Operadores:** ^ , * , / , AND , + , - , OR , < , ≥ , ≤ , = , > , < , DIV , MOD , XOR , SHL , SHR
- **Gestión de errores:** TRY...EXCEPT , TRY...FINALLY
- **Otros:** EXIT , constructores de matrices [1, 2, 3] , acceso a objetos ObjectName.Property

Instrucciones If

Existen dos formas: IF...THEN...END IF e IF...THEN...ELSE...END IF . Si la expresión es verdadera, se ejecutan las instrucciones de THEN . Si es falsa, se ejecutan las instrucciones de ELSE (si las hubiera).

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

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

Si la instrucción ocupa una sola línea, no necesita `END IF` :

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

Instrucciones `Select Case`

Si la `selectorExpression` coincide con una de las expresiones `caseexpr` , se ejecutarán las instrucciones correspondientes. De lo contrario, se ejecutará la instrucción `CASE ELSE` .

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

Instrucciones `For`

La instrucción `FOR` repite la ejecución de instrucciones hasta que un contador alcanza un valor final.

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

La parte `STEP` es opcional; si se omite, el valor del paso es 1.

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

Instrucciones `While`

Una instrucción `WHILE` repite las instrucciones mientras una condición de control sea verdadera. La condición se evalúa antes de ejecutar las instrucciones.

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

Instrucciones Do... Loop

Las instrucciones se ejecutan **WHILE** (mientras) una expresión sea verdadera, o **UNTIL** (hasta que) una expresión sea verdadera. La condición se puede comprobar antes o después de la iteración.

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

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

Declaraciones de funciones y subs

Las declaraciones son similares a las del Basic estándar. Para devolver un valor desde una función, asígnelo a la variable implícita con el mismo nombre que la función, o use la instrucción **Return**. Los parámetros se pueden pasar por referencia mediante la directiva **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
```

Los subs y las funciones son públicos de forma predeterminada, pero se pueden declarar como **PRIVATE**.

Recetario de la API de VBA

Este tema ofrece una colección completa de fragmentos de código VBA ("recetas") para un control avanzado de su aplicación protegida en tiempo de ejecución. Con la API VBA de XLS Padlock, puede recuperar información del sistema, gestionar el guardado y la carga, personalizar la interfaz de usuario e interactuar con funciones en línea.

Los dos objetos principales de la API son:, `Application.COMAddIns("GXLSForm.GXLSFormula").Object` para la mayor parte de la información en tiempo de ejecución.,

`Application.COMAddIns("GXLS.GXLSPLock").Object` para las operaciones relacionadas con el guardado.

Cada receta a continuación proporciona una función o subrutina VBA lista para usar.

Información de la aplicación y del sistema

Estas recetas le ayudan a recuperar información sobre el entorno y el estado de la aplicación.

Comprobar si el libro está protegido

Utilice esto para confirmar que su código se está ejecutando dentro de una aplicación protegida creada por 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
```

Obtener el nombre del archivo EXE de la aplicación

Recupera el nombre del archivo `.exe` en ejecución.

```
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
```

Obtener la versión del archivo EXE y del producto

Recupera las cadenas de versión que definió en la sección "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
```

Uso: `ReadVersionInfo("ProductVersion")` devuelve la versión del producto (Product Version)., `ReadVersionInfo("FileVersion")` devuelve la versión del archivo (File Version).

Obtener el ID del sistema del usuario

Recupera el ID del sistema único que se requiere para las claves de activación bloqueadas por hardware. Este es el ID que le enviarán sus clientes.

```
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
```

Obtener parámetros de la línea de comandos

Recupera los parámetros de la línea de comandos pasados al archivo `.exe` de su aplicación.

```
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
```

Uso: `ReadParamStr(1)` devuelve el primer parámetro, `ReadParamStr(2)` el segundo, y así sucesivamente.

Guardar y cargar libros

Gestione los archivos de guardado seguros (`.xlsc` o `.xlsce`) mediante programación.

Obtener la ruta del archivo de guardado actual

Recupera la ruta completa al archivo de guardado seguro que está cargado actualmente.

```
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
```

Obtener la ruta de la carpeta de guardado local

Recupera la ruta a la carpeta local donde XLS Padlock almacena la configuración de su aplicación y los archivos de guardado seguros.

```
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
```

Ejecutar código VBA después de guardar

XLS Padlock puede llamar automáticamente a la subrutina `XLSPadlock_OnAfterSave` después de que un usuario guarde su trabajo. Esto resulta útil para el registro, para mostrar un mensaje de confirmación o para otras acciones posteriores al guardado.

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

Abrir un archivo de guardado existente con VBA

Abra un archivo de guardado seguro mediante programación. Tenga en cuenta que esto cargará el archivo en una **nueva instancia de 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
```

Sugerir un nombre de archivo para el cuadro de diálogo de guardado

Establezca el nombre de archivo predeterminado que aparece en el cuadro de diálogo "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")
```

Guardar una copia segura sin solicitar confirmación

Guarde una copia segura del libro directamente en una ruta de archivo especificada sin mostrar el cuadro de diálogo "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
```

Migrar datos de usuario de una versión anterior

Si publica una nueva versión de su libro, es posible que necesite importar datos de un archivo de guardado creado con una versión anterior.

→ [Aprenda a migrar datos de usuario con VBA](#)

Interfaz de usuario

Personalice elementos de la interfaz, como los cuadros de diálogo.

Ocultar mediante programación los cuadros de diálogo de espera/carga

Puede ocultar el cuadro de diálogo "Loading workbook, please wait..." con una llamada VBA en lugar de esperar a que se cierre automáticamente.

```
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
```

Activación y validación en línea

Interactúe con las funciones en línea de XLS Padlock.

Comprobar si hay conexión a Internet

Comprueba si hay una conexión a Internet activa disponible en el equipo del usuario.

```
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
```

Obtener el token de activación en línea

Recupera un hash del token de activación devuelto por el servidor web tras una activación en línea correcta.

```
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
```

Comprobar si la validación en línea se ha realizado correctamente

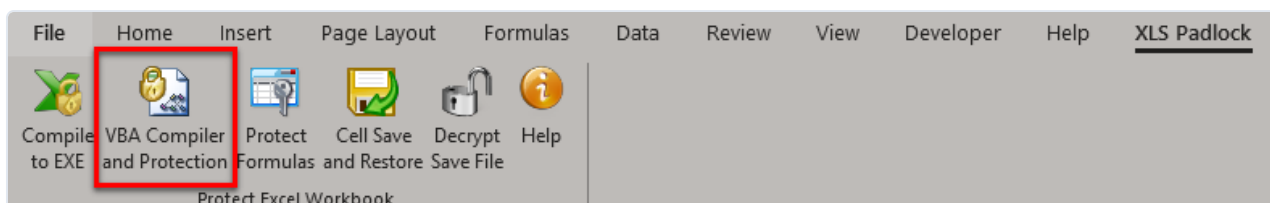
Obtiene el resultado del último intento de validación en línea.

```
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
```

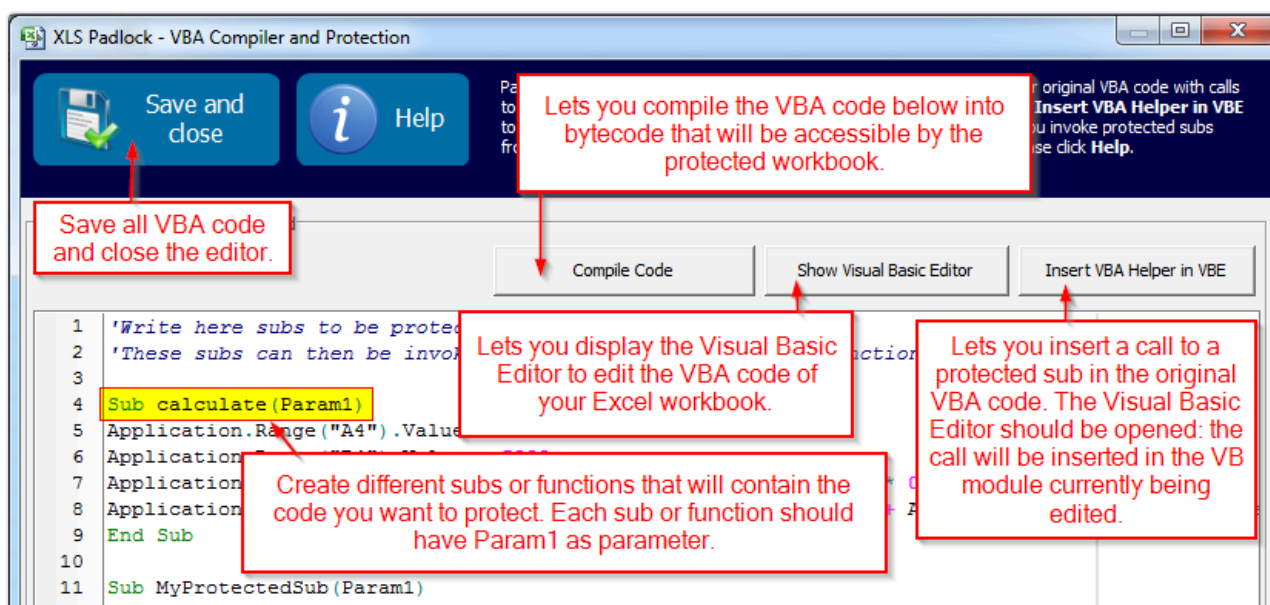
Escribir y compilar VBA seguro

Puede **proteger las partes sensibles de su código VBA compilándolas en bytecode con XLS Padlock**. Recomendamos combinar el bytecode compilado con su código VBA existente para lograr un enfoque equilibrado entre seguridad y desarrollo.

1. En Excel, haga clic en "VBA Compiler and Protection" para abrir el editor y compilador VBA de XLS Padlock:



Se abrirá el editor VBA de XLS Padlock:



2. En el área de texto, introduzca el código VBA que desea compilar. Puede agrupar su código en `subs` y `functions`. Para simplificar este proceso, puede abrir el Editor de Visual Basic haciendo clic en "Show Visual Basic Editor" y copiar y pegar su código.

De forma predeterminada, todos los subs y funciones tienen un parámetro llamado `Param1`. No obstante, puede utilizar más parámetros.

3. Cuando haya terminado, haga clic en "**Compile Code**". El código VBA se compila instantáneamente en bytecode, que XLS Padlock almacenará hasta que compile el archivo EXE de su aplicación.

CONSEJO

Si se detecta un error, se mostrará y el código correspondiente quedará subrayado en rojo.

1. Haga clic en "Save and close" para cerrar el editor.

👉 Después de compilar su código, deberá **modificar sus macros VBA originales para que llamen a la versión compilada**. Aprenda a hacerlo en la guía [Invocar código VBA compilado en tiempo de ejecución](#).

Invocar VBA compilado en tiempo de ejecución

Para ejecutar su [código VBA compilado](#), debe llamarlo desde el código VBA normal de su libro mediante una función auxiliar.

Primero, abra el Editor de Visual Basic y pegue la siguiente función auxiliar en un módulo de su libro:

```
' 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
```

Inserción automática

Esta función se puede insertar automáticamente en su módulo VB activo haciendo clic en el botón "Insert VBA Helper in VBE" del editor VBA de XLS Padlock.

La función auxiliar `CallXLSPadlockVBA` toma dos parámetros:

- **ID**: El nombre del `Sub` o `Function` compilado que desea invocar.
- **Param1**: Un parámetro opcional que desea pasar al código compilado.

Ejemplo

Imagine que su módulo VBA original contiene el siguiente código:

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

Después de trasladar este código al VBA Compiler, reemplazaría el sub original por una llamada a la función auxiliar:

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

Aquí indicamos el nombre del sub protegido, `Calculate`, y pasamos una cadena vacía como segundo parámetro porque en este caso no se utiliza. La variable `res` contendrá el resultado de la llamada, lo cual resulta útil sobre todo si el código invocado es una `Function` que devuelve un valor.

Ejecución en tiempo de diseño

Una de las características más potentes de XLS Padlock es que esta configuración funciona incluso en tiempo de diseño. Si ejecuta el sub `Calculate()` desde el VBE antes de compilar el libro, XLS Padlock ejecutará el código protegido, lo que le permite probar su lógica sin tener que compilar primero la aplicación.

👉 Vea también: Aprenda a [pasar más de un parámetro a su código compilado](#chapter-passing-more-parameters-to-the-compiled-vba-code).

Acceder a los objetos de Excel

Al escribir código para el compilador VBA, debe usar explícitamente el objeto `Application` para acceder al modelo de objetos de Excel.

Para una referencia detallada de las propiedades y los métodos disponibles, consulte la [documentación del Microsoft Office Dev Center](#).

Ejemplos

Referenciar un rango (Range)

Para referenciar una celda o un rango de celdas en su hoja de cálculo, debe anteponer `Application` al objeto `Range`.

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

Usar funciones de hoja de cálculo

Para usar las funciones de hoja de cálculo de Excel, debe usar el objeto

`Application.WorksheetFunction`.

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

Las funciones requieren todos los parámetros

Al llamar a funciones integradas de Excel como `InputBox` desde código compilado, **debe especificar todos los parámetros**, ya que los parámetros predeterminados no son compatibles.

Ejemplo completo: convertir una macro

Aquí tiene una macro VBA estándar:

```
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
```

Aquí tiene el mismo código modificado para el compilador VBA de XLS Padlock. Observe la adición del objeto `Application` y los parámetros explícitos para `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
```

Su módulo VBA original se modificaría entonces para llamar a la versión compilada:

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

Pasar matrices

Puede pasar distintos tipos de variables al código VBA compilado, incluidos los arrays estáticos.

Suponga que tiene la siguiente función en el compilador VBA:

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

En su módulo VBA normal de Excel, puede llamar a esta función y pasar un array.

Advertencia

El array debe definirse como 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
```

Pasar más parámetros

De forma predeterminada, la función auxiliar utilizada para ejecutar código VBA compilado admite solo un parámetro.

```
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
```

El método `PLEvalVBA` del objeto `XLSPadlock` toma dos parámetros: el **ID** del sub/función compilado y un único parámetro **Param1**.

Para pasar más parámetros, el objeto `XLSPadlock` proporciona métodos adicionales:

- `XLSPadlock.PLEvalVBA2(ID, Param1, Param2)` para dos parámetros.
- `XLSPadlock.PLEvalVBA3(ID, Param1, Param2, Param3)` para tres parámetros.

También es posible [pasar más parámetros usando arrays](#).

Por ejemplo, una función auxiliar para pasar dos parámetros tendría este aspecto:

```
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
```

Gestionar errores

Pueden producirse errores en tiempo de ejecución cuando se ejecuta su código VBA compilado. Mientras que el VBA estándar utiliza la instrucción `On Error` para la gestión de errores, el VBA compiler de XLS Padlock utiliza un bloque `Try...Except`.

Cuando se produce un error dentro del bloque `Try` (o en cualquier procedimiento que este llame), el compilador saltará inmediatamente al bloque `Except` para su gestión.

Sintaxis

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

CONSEJO

Un error no gestionado será mostrado por el VBA compiler en tiempo de ejecución, [a menos que desactive esta opción.](#)

Ejemplo de código

```
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
```

Error OLE 800A03EC

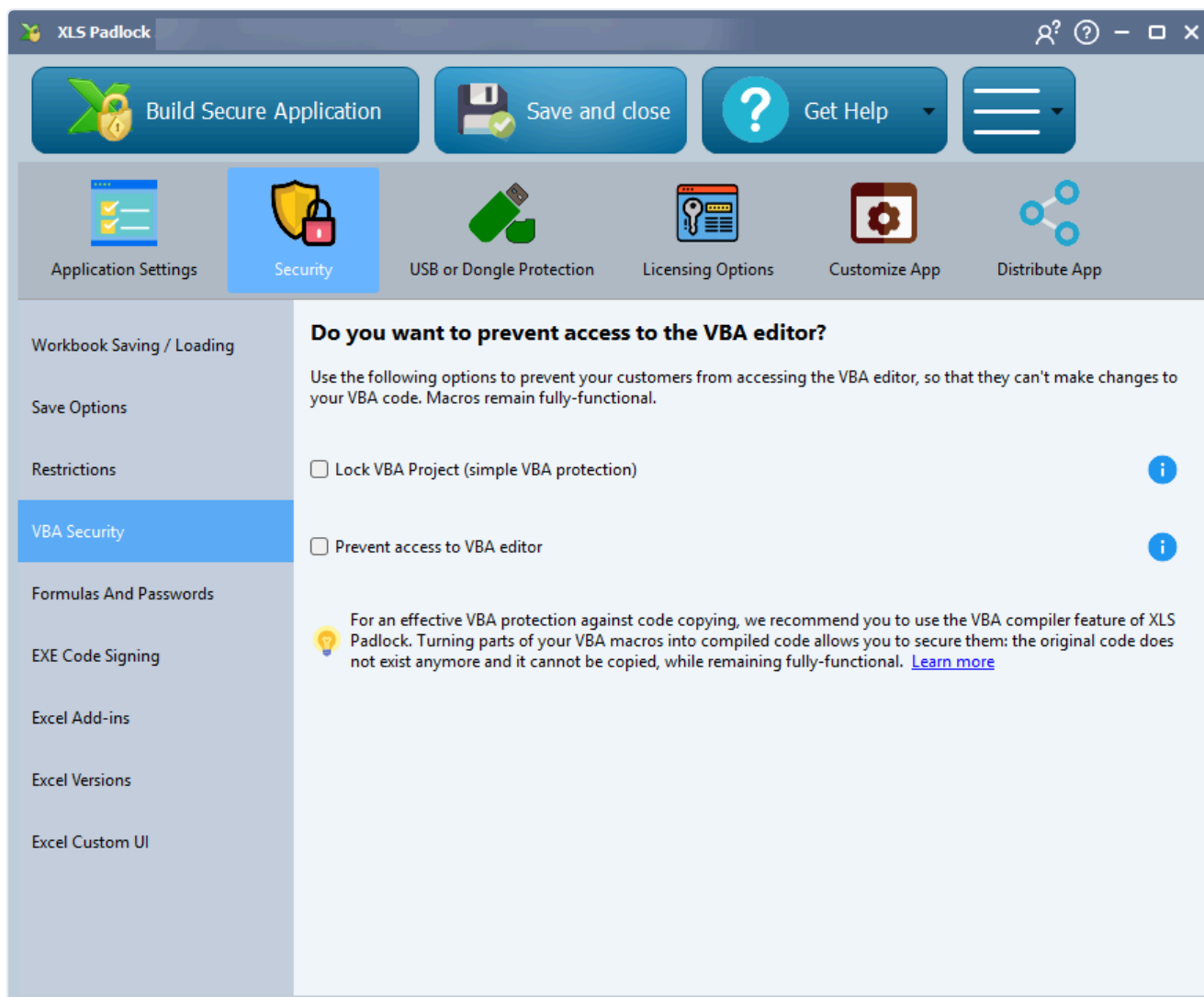
Para resolver este error, asegúrese de que la opción **“Trust access to the VBA project object model”** esté activada en la configuración del Centro de confianza (Trust Center) de Excel.

Para obtener instrucciones sobre cómo encontrar esta opción, consulte la página oficial de soporte de Microsoft Office:

[Activar o desactivar macros en documentos de Office](#)

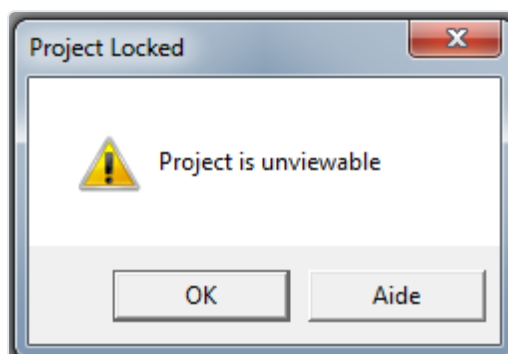
Protección del código VBA

Su código VBA es un activo valioso. XLS Padlock proporciona varias capas de protección para protegerlo frente a la manipulación o el robo. A continuación se presentan dos opciones de seguridad fundamentales, seguidas de nuestra solución más avanzada para una protección máxima.



Opción 1: Bloquear su proyecto VBA

El paso más sencillo es [bloquear su proyecto VBA](#). Esto hace que su código no se pueda ver en el Editor de Visual Basic. Si un usuario intenta acceder al proyecto, encontrará un mensaje de error, lo que bloquea de forma efectiva la inspección casual de su código.



Opción 2: Impedir el acceso al editor de VBA

Para una capa de seguridad adicional, puede [impedir que el editor de VBA se abra por completo](#). XLS Padlock vigilará de forma continua y cerrará automáticamente cualquier intento de abrir el VBE, garantizando que incluso los usuarios expertos queden bloqueados.

Protección definitiva: compilación de VBA

Para el máximo nivel de seguridad, XLS Padlock le permite **compilar su código VBA en código binario nativo**. A diferencia de la simple ofuscación, que solo hace que el código sea difícil de leer, la compilación transforma su lógica en un formato no legible por humanos que es casi imposible de aplicar ingeniería inversa. Su código VBA original se elimina del libro y se reemplaza por llamadas al código compilado y seguro dentro del EXE. Esta es la defensa definitiva para su propiedad intelectual.

→ [Más información sobre nuestro potente VBA Compiler](#)

Ocultar y bloquear el código VBA

XLS Padlock ofrece una función dedicada [Lock VBA Project](#). Sin embargo, si decide no utilizarla, todavía puede recurrir a la protección nativa de Excel. XLS Padlock hace que este método sea más seguro.

Cómo bloquear su proyecto VBA en Excel

1. En Excel, abra el Visual Basic Editor (VBE) pulsando `Alt+F11` o seleccionándolo desde la pestaña `Developer`.
2. En el VBE, vaya al menú **Tools** y seleccione **VBAProject Properties...**
3. En el cuadro de diálogo, vaya a la pestaña **Protection**.
4. Marque la casilla **Lock project for viewing** e introduzca una contraseña segura.

Una vez que el libro se guarde y se vuelva a abrir, los usuarios no podrán ver ni editar el código VBA sin la contraseña.

Por qué esto es más seguro con XLS Padlock

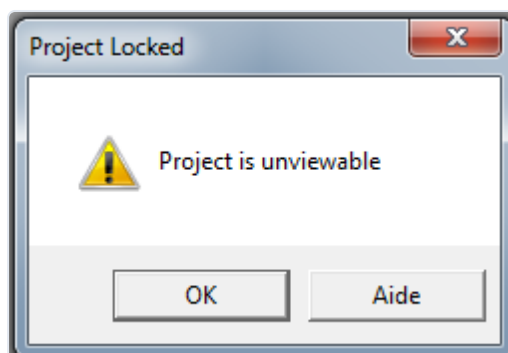
Normalmente, la protección por contraseña del VBA de Excel se considera débil porque existen muchas herramientas capaces de eliminar la contraseña. Estas herramientas funcionan accediendo directamente al archivo de Excel.

Sin embargo, cuando su libro se compila en un EXE con XLS Padlock, el archivo `.xls` o `.xlsm` original se cifra y se incrusta dentro de la aplicación. Las herramientas para descifrar contraseñas no pueden acceder a este archivo protegido, lo que las vuelve inútiles. Esta combinación proporciona una defensa mucho más sólida para su código VBA.

Bloquear el proyecto VBA

La opción **Lock VBA Project** ofrece una forma sencilla de impedir que los usuarios finales accedan a su proyecto VBA. Esta función marca el proyecto VBA como bloqueado, lo que hace imposible verlo, accederlo o modificarlo. No utiliza protección por contraseña.

Si el usuario final intenta acceder a un proyecto VBA bloqueado, se muestra el siguiente mensaje de error (Project Locked, Project is unviewable):



Elimine las contraseñas VBA existentes

Si utiliza esta opción, le recomendamos que elimine cualquier contraseña existente de su proyecto VBA. Una contraseña se vuelve redundante cuando el proyecto está bloqueado. Además, usar una contraseña larga con esta opción puede provocar bloqueos en Excel 2007.

Si prefiere conservar una contraseña, considere usar en su lugar la opción [Prevent access to VBA editor] (#chapter-forbid-access-to-the-vba-editor-vbe).

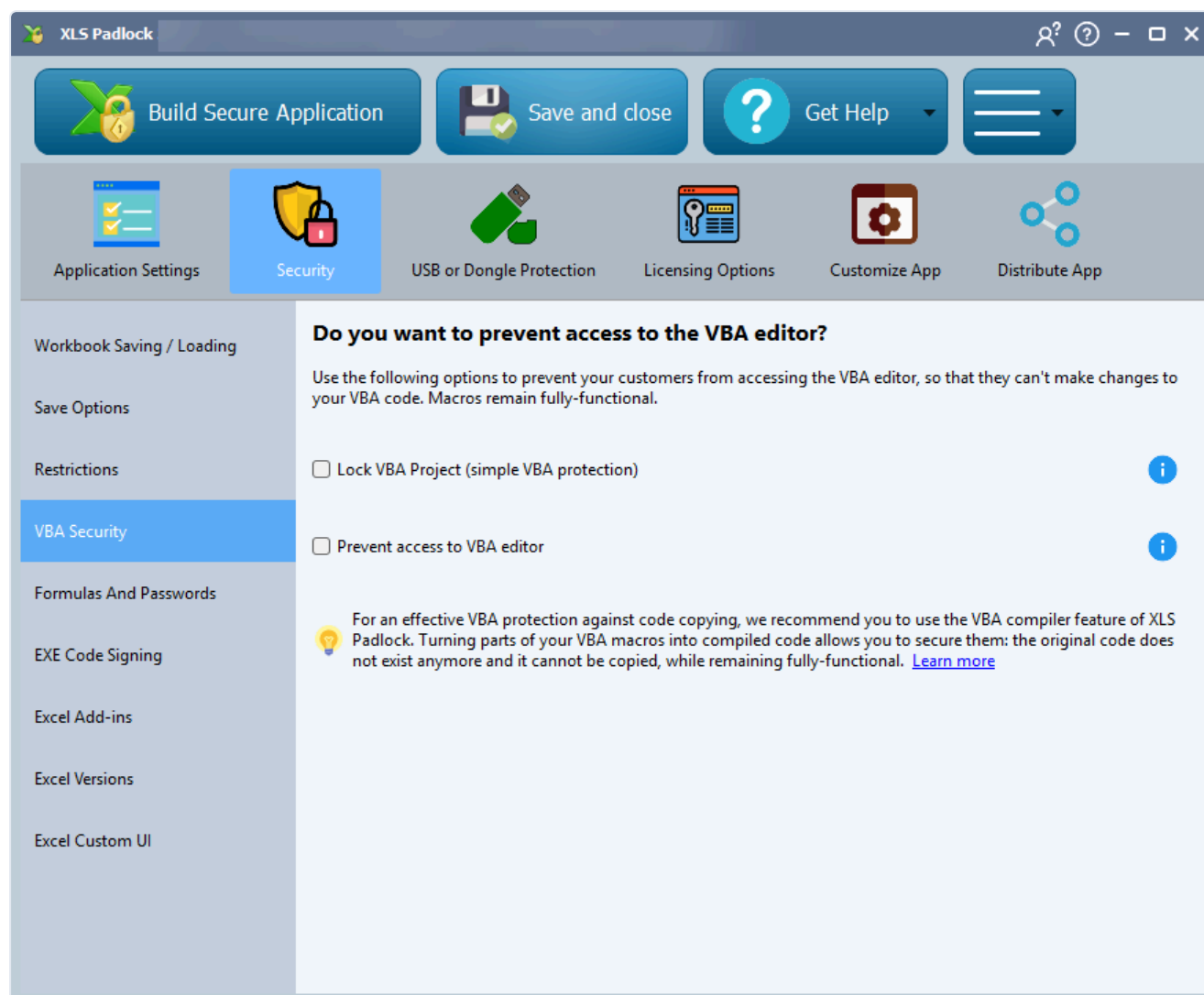
Tenga en cuenta que las macros siguen siendo funcionales.

Esta opción es compatible con nuestro [compilador VBA](#), que ofrece un nivel de seguridad superior para su código VBA.

Prohibir el acceso al VBE

XLS Padlock le ofrece opciones para **impedir que sus clientes accedan al editor VBA (VBE)**, evitando así de forma efectiva que vean o alteren su código VBA, al tiempo que garantiza que sus macros sigan siendo plenamente funcionales.

Estas opciones están disponibles en la pestaña "Security" de XLS Padlock.



Lock VBA Project (protección VBA simple)

Esta opción bloquea su proyecto VBA mediante programación para dificultar el acceso. Aunque ofrece un nivel básico de protección, puede ser eludida por algunas herramientas.

Prevent Access to VBA Editor

Esta es una medida de seguridad más sólida. XLS Padlock realizará una comprobación periódica y, si un usuario intenta abrir el Visual Basic Editor (VBE), XLS Padlock cerrará de inmediato la ventana del VBE. Esto les impide incluso intentar ver el código.

Esta opción es compatible con la opción "Lock VBA Project" y con nuestro potente [compilador VBA](#).

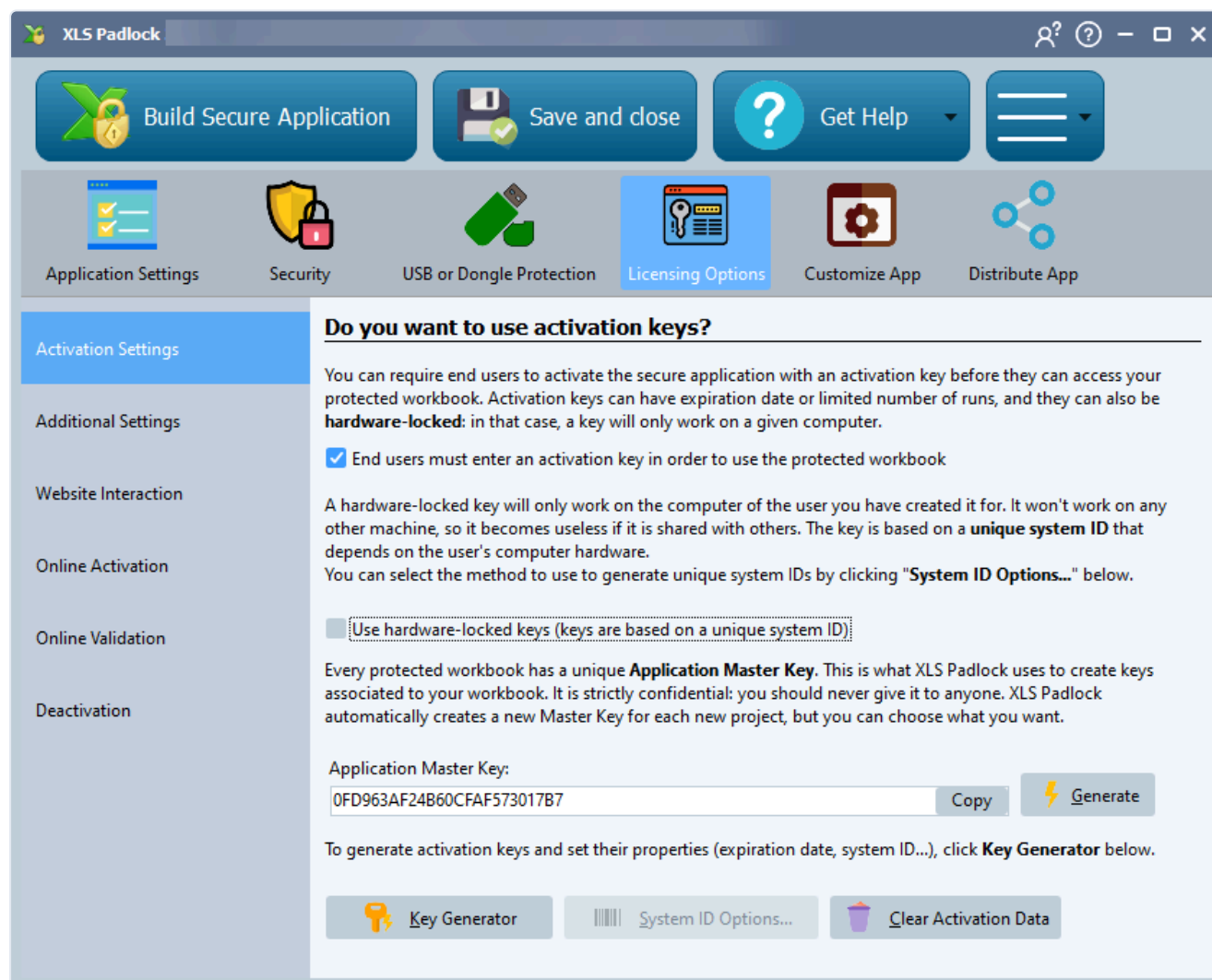
Para la máxima protección del código VBA

👉 Para una protección sólida contra el robo de código y la ingeniería inversa, recomendamos encarecidamente utilizar la [función de compilador VBA](#).

Al compilar partes de sus macros VBA, el código fuente original se elimina y se sustituye por código binario. Este código compilado es extremadamente difícil de someter a ingeniería inversa, lo que garantiza que sus algoritmos y su lógica de negocio queden protegidos. Es la forma más eficaz de proteger su propiedad intelectual VBA.

Guía de activación y licencias

Esta guía cubre las distintas opciones relacionadas con las **claves de activación**, las **licencias**, la **validación en línea** y la **desactivación** para sus aplicaciones protegidas.



Conceptos básicos

Habilitación de las claves de activación

Para exigir que los usuarios dispongan de una clave para usar su aplicación, habilite la opción: **"End users must enter an activation key in order to use the protected workbook"**. Este es el interruptor principal para activar las funciones de licencia.

Application Master Key

Cada libro protegido tiene una **Application Master Key** única. XLS Padlock usa esta clave para generar las claves de activación asociadas a su libro. Es **estrictamente confidencial** y nunca debe compartirse.

XLS Padlock crea automáticamente una nueva Master Key para cada proyecto.

Application GUID y Secret Key

- El **Application GUID** es un identificador único que la aplicación usa para almacenar sus ajustes y gestionar los archivos guardados en el equipo del usuario.
- La **Application Secret Key** se usa para cifrar y proteger los archivos guardados por el usuario (`.XLSC` o `.XLSCE`). Garantiza que los archivos guardados con su aplicación solo puedan abrirse con su aplicación.

IMPORTANTE

Si cambia el Application GUID o la Secret Key, los archivos guardados anteriormente dejarán de ser accesibles para sus usuarios.

Publicar una actualización que deba invalidar los archivos guardados antiguos de los usuarios

Para forzar que una nueva versión ignore cualquier archivo guardado con versiones anteriores, por ejemplo, cuando la estructura del libro ha cambiado lo suficiente como para que los archivos guardados antiguos no deban recargarse, genere una **nueva Application Secret Key** antes de recompilar, pero **deje el Application GUID sin cambios** para que las claves de activación existentes sigan siendo válidas. En el primer arranque del usuario, el archivo guardado antiguo se detectará como incompatible y se le pedirá que cargue el libro original incrustado en su lugar. Consulte [Actualizaciones del libro](#) para conocer el flujo completo de actualización.

CONSEJO

El estado de activación (prueba / registrado) se puede obtener mediante código VBA.

Permitir que el usuario final cambie la clave de activación

Un usuario puede cambiar su clave de activación ejecutando la aplicación con el [modificador de línea de comandos](#) `-enterkey` (por ejemplo, `MYAPP.EXE -enterkey`). Esto resulta útil para reemplazar una clave que está a punto de caducar.

Borrar los datos de activación

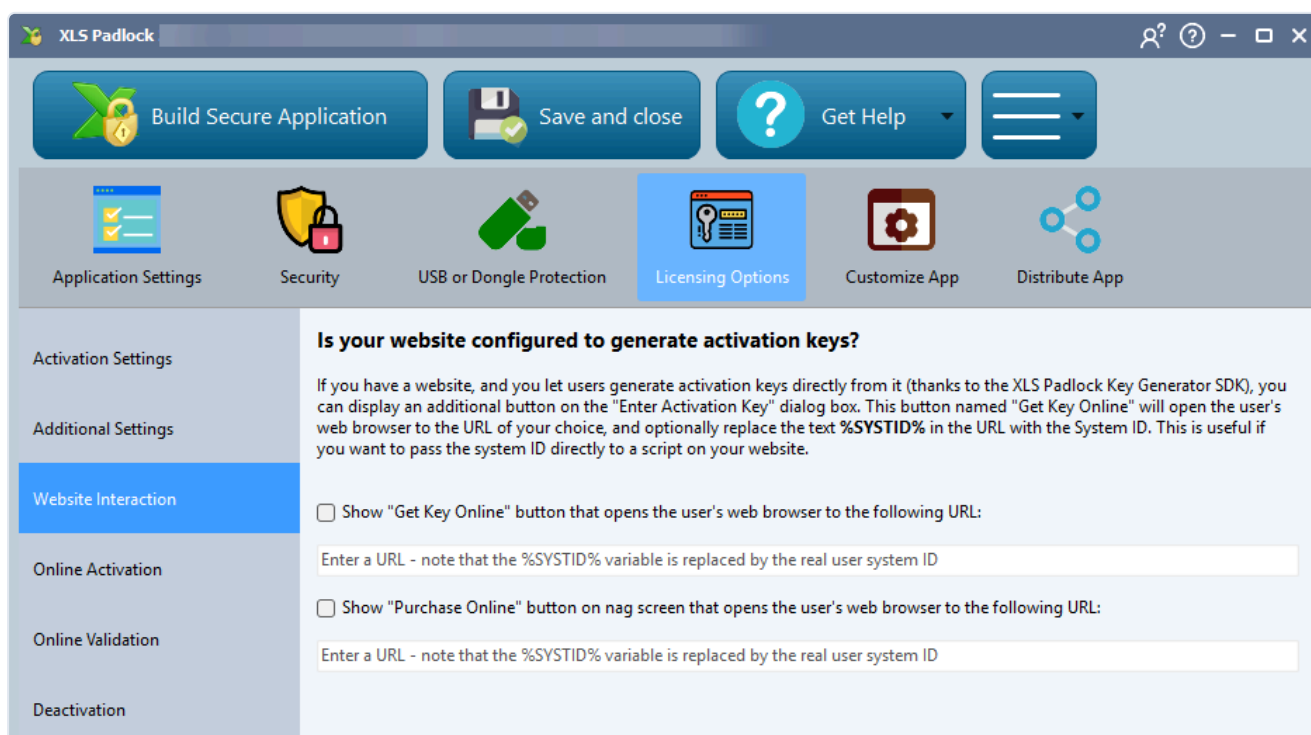
Con fines de prueba en su propio equipo, el botón "**Clear Activation Data**" del Key Generator restablece toda la información de activación almacenada localmente.

Claves bloqueadas por hardware

Una clave bloqueada por hardware **solo funcionará en el equipo concreto** para el que se generó, lo que la hace inservible si se comparte. La clave se basa en un **System ID único** derivado del hardware del equipo del usuario.

- **Opciones de System ID:** puede seleccionar qué componentes de hardware (CPU, dirección MAC, etc.) se utilizan para generar el System ID. Sea coherente con estas opciones después de implementar su aplicación.
- **Obtención del System ID:** cuando un usuario ejecuta la aplicación, se le mostrará su System ID, que debe enviarle a usted. A continuación, usted utiliza este ID en el Key Generator para crear su clave.
- **Automatización en línea:** para evitar el intercambio manual de System IDs, puede usar las funciones de [activación en línea](#).

Interacción con el sitio web



Botón Get Key Online

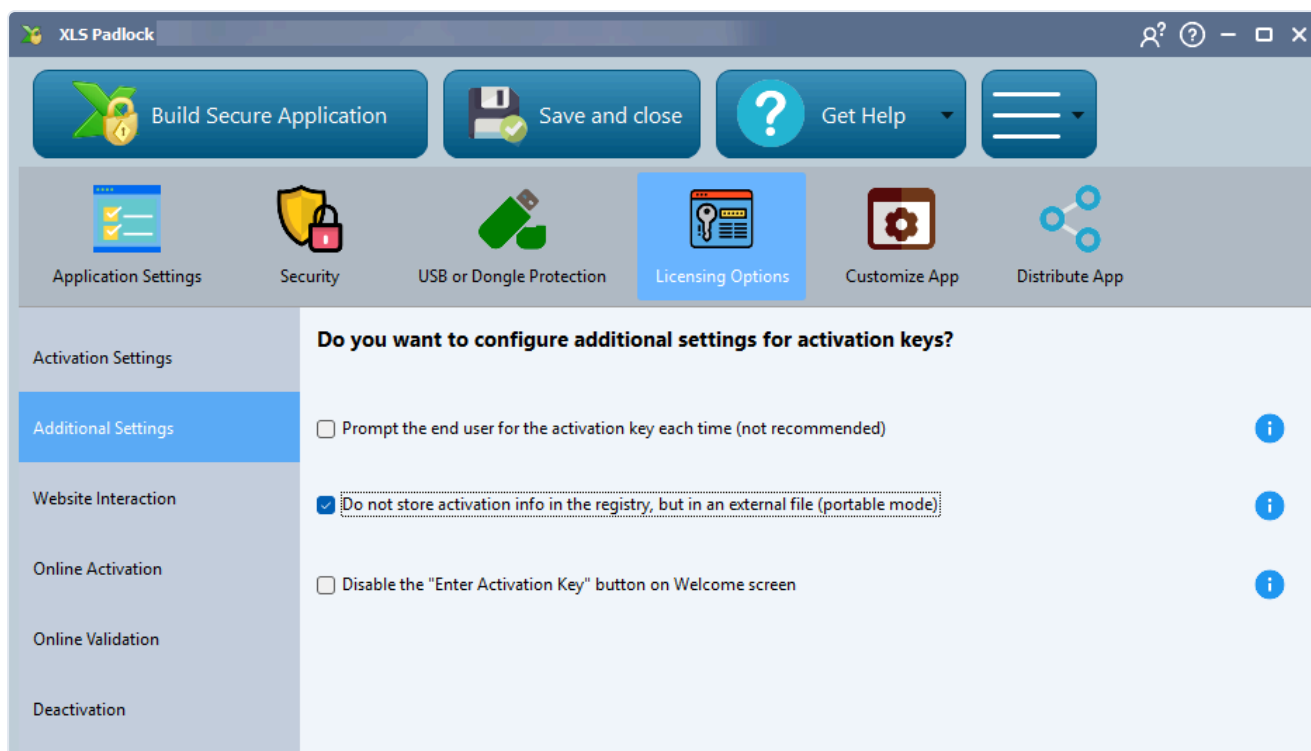
Puede mostrar un botón "Get Key Online" en la ventana de activación. Este botón abre el navegador web del usuario en una URL que usted especifique. Puede usar el marcador de posición `%SYSTID%` en la URL para pasar automáticamente el System ID del usuario al script de generación de claves de su sitio web.

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

Mostrar el botón "Purchase Online"

Para las versiones de prueba con una pantalla recordatoria, puede añadir un botón "Purchase Online" que dirija al usuario a su tienda o página de compra.

Ajustes adicionales



Modo portátil

Habilite la opción **"Do not store activation info in the registry, but in an external file (portable mode)"**. Esto crea un archivo `.lic` oculto en la misma carpeta que su EXE, lo que permite ejecutar la aplicación desde una unidad USB sin escribir en el registro de Windows.

IMPORTANTE

Este modo no se recomienda para las versiones de prueba, ya que eliminar el archivo `.lic` restablecerá el periodo de prueba.

Pedir al usuario final la clave de activación cada vez

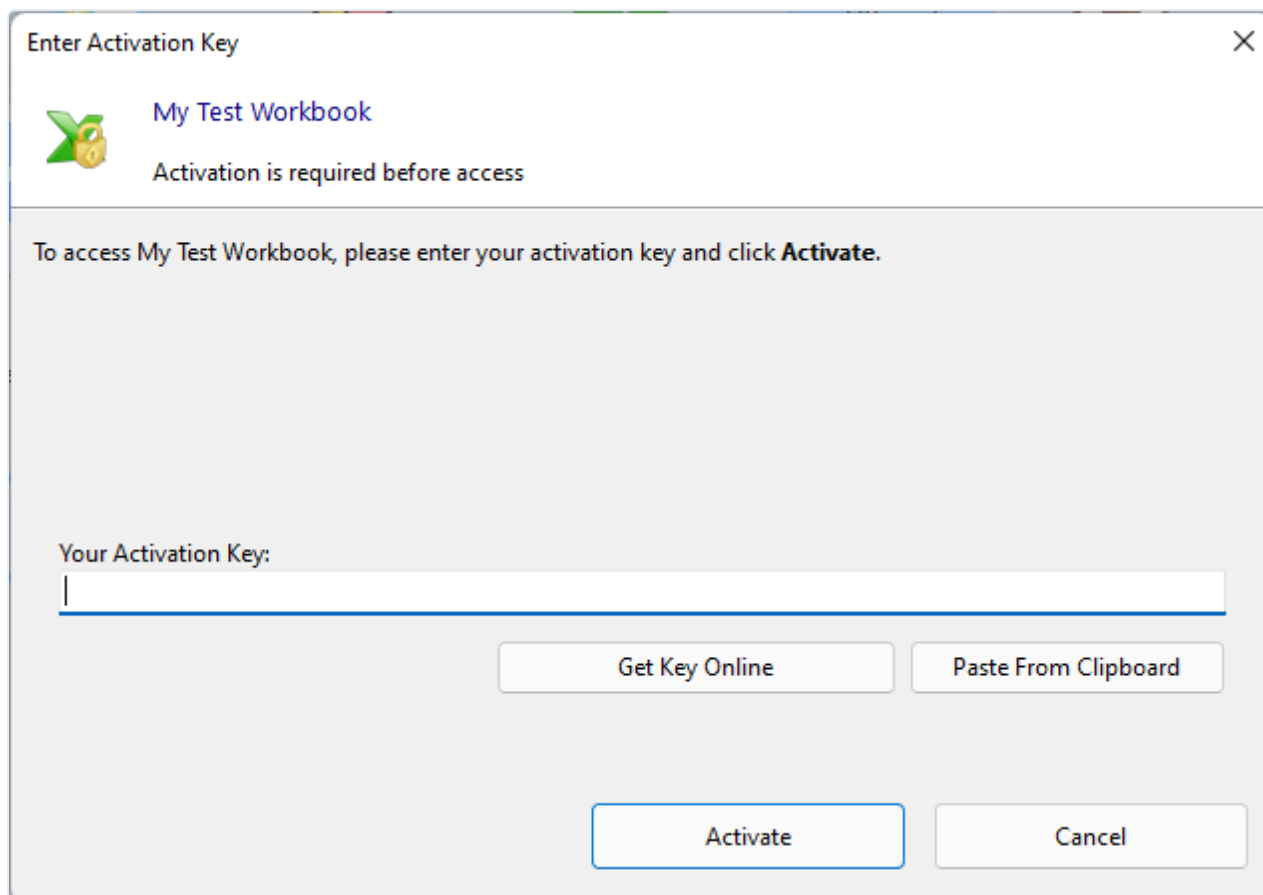
Al habilitar "Prompt the end user for the activation key each time", sus usuarios tendrán que introducir la clave de activación cada vez que abran su aplicación. No se recomienda, porque resulta fastidioso para su cliente y no puede establecer funciones de caducidad para la clave.

Deshabilitar el botón "Enter Activation Key" en la pantalla de bienvenida

Esto eliminará la posibilidad de que sus usuarios finales introduzcan una clave de activación cuando abran su aplicación de libro de Excel. No obstante, no se recomienda usar esta opción.

Claves de activación

Las claves de activación son una función esencial de XLS Padlock, que ofrece un **potente sistema de licencias para sus libros de Excel**. Este sistema le permite exigir que los usuarios finales activen la aplicación con una clave antes de poder acceder a su libro protegido:

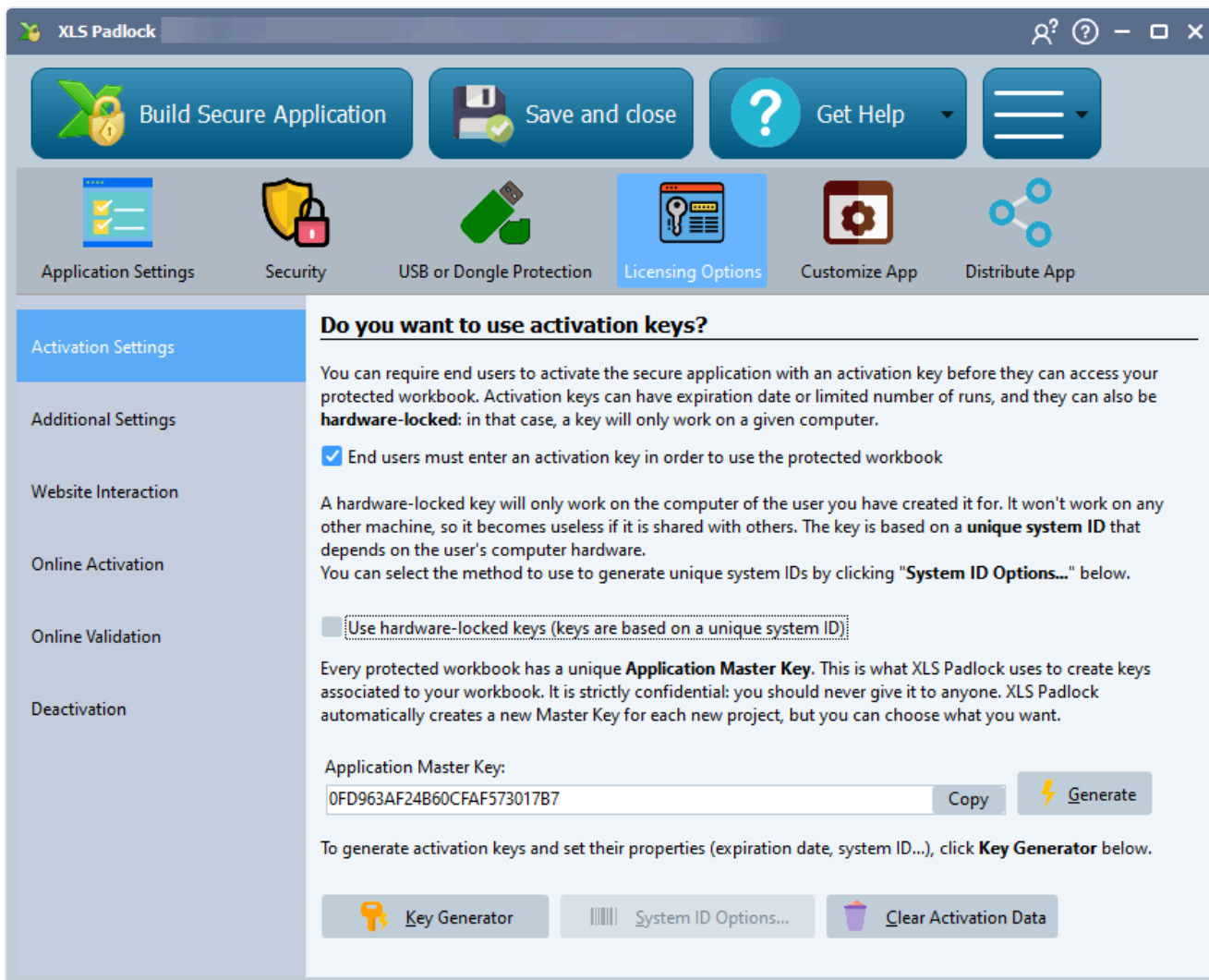


👉 Las claves de activación se pueden configurar con una [fecha de caducidad](#) o un [número limitado de ejecuciones](#). También se pueden [bloquear por hardware](#), lo que significa que una clave solo funcionará en un equipo concreto.

CONSEJO

Además del [Key Generator](#) integrado, ofrecemos un generador de claves independiente capaz de crear miles de claves a la vez, así como un SDK de generación de claves para la integración con su servidor o sitio web.

Por último, puede implementar funciones de [activación en línea](#) y [validación](#) para controlar de forma remota cómo acceden los usuarios a su libro y automatizar la entrega de las claves de activación.



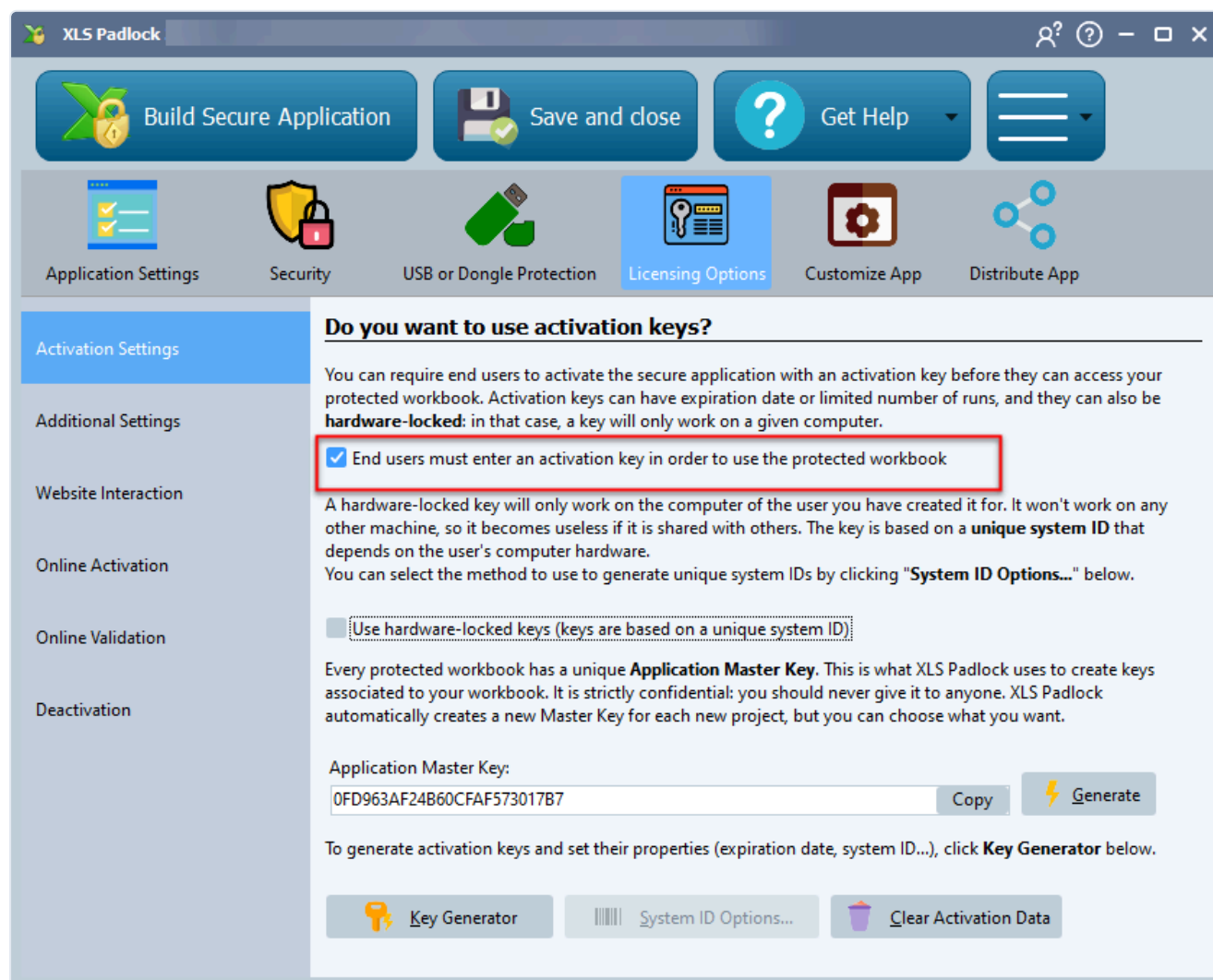
👉 Para empezar, consulte nuestro tutorial sobre [cómo configurar las claves de activación](#).

Configurar claves de activación

Las **claves de activación** son un potente sistema de licencias que le permite controlar quién puede acceder a sus libros protegidos. Esta guía le muestra cómo empezar.

1. Habilitar las claves de activación

En los ajustes de seguridad de XLS Padlock, marque la casilla **"End users must enter an activation key in order to use the protected workbook"**.

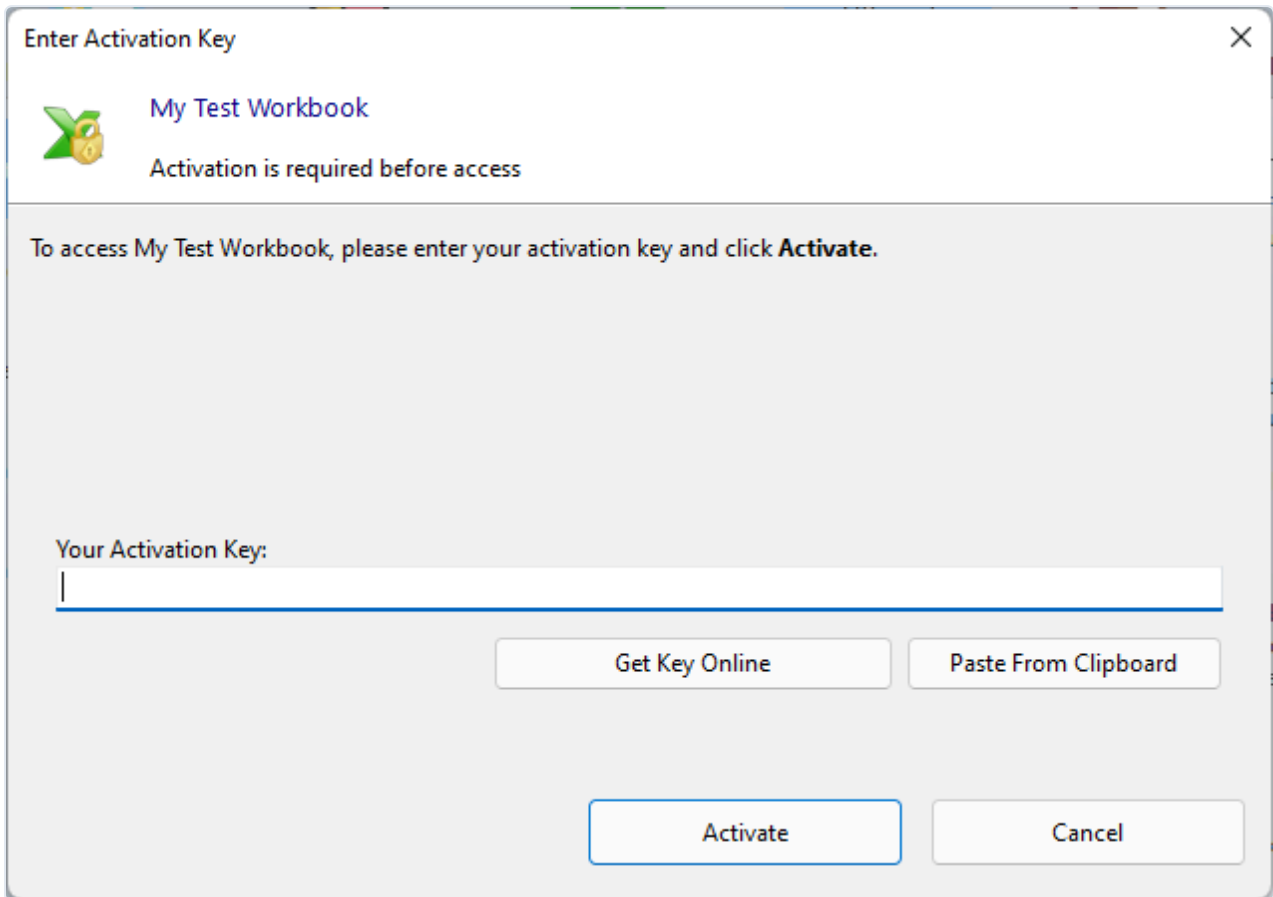


The screenshot shows the XLS Padlock application window with the 'Licensing Options' tab selected. The 'End users must enter an activation key in order to use the protected workbook' checkbox is checked and highlighted with a red box. Below it, there is a section for 'Use hardware-locked keys (keys are based on a unique system ID)' and an 'Application Master Key' field with a 'Generate' button.

Una vez hecho esto, recompile su aplicación.

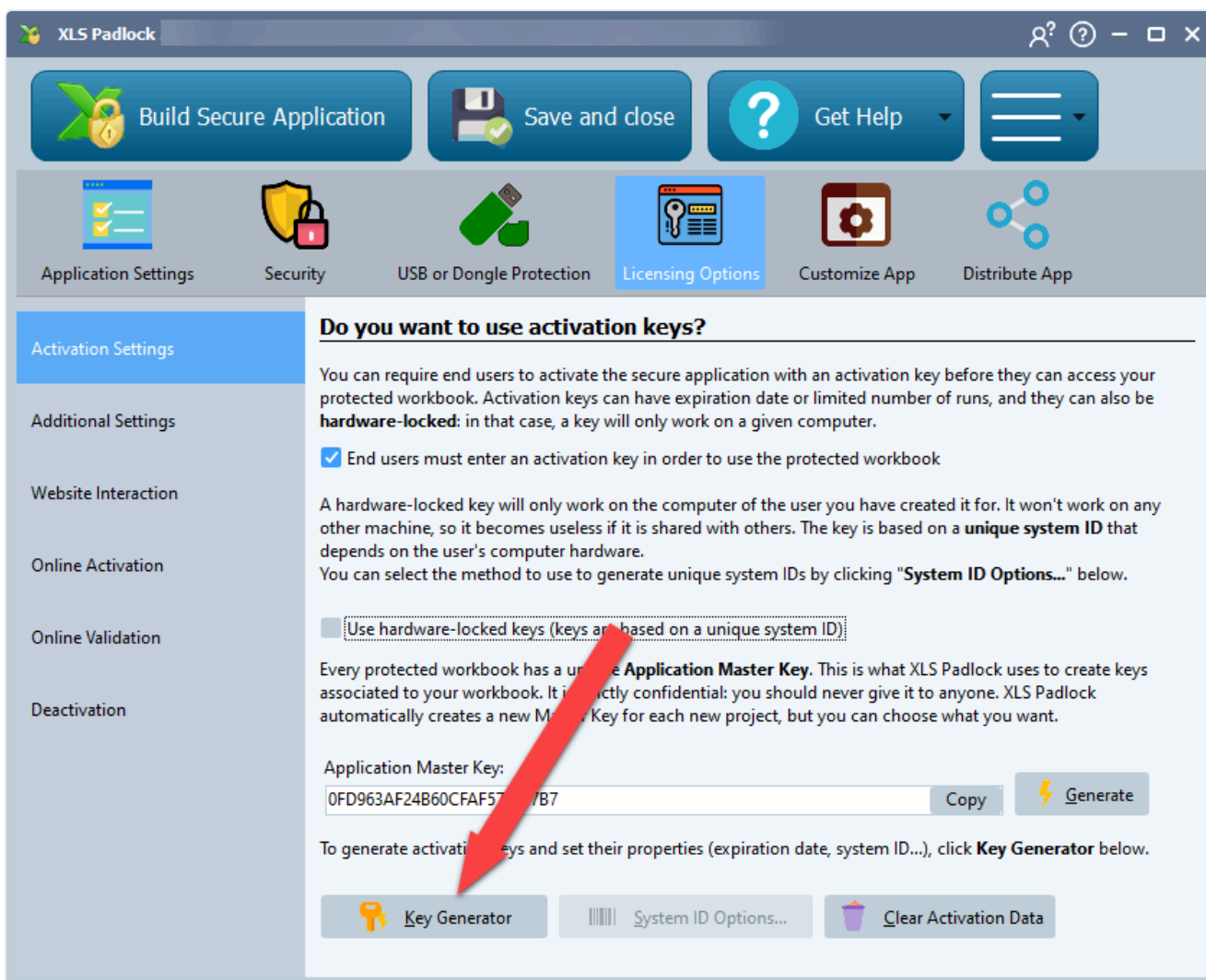
2. Probar su aplicación

Cuando ejecute la aplicación recién compilada, ahora se le pedirá que introduzca una clave de activación antes de poder acceder al libro:



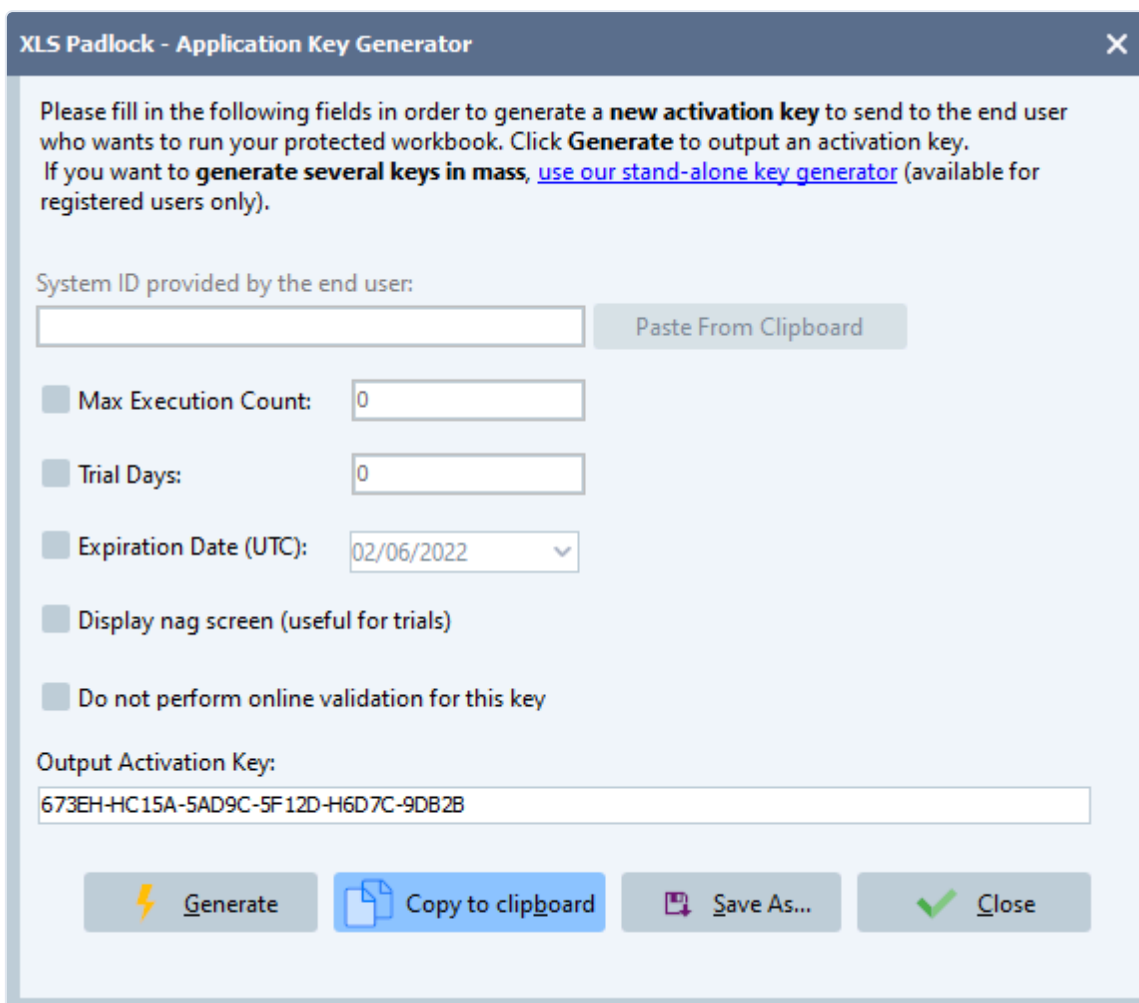
3. Generar las claves de activación

Para crear claves de activación, abra el [Key Generator](#) en XLS Padlock.

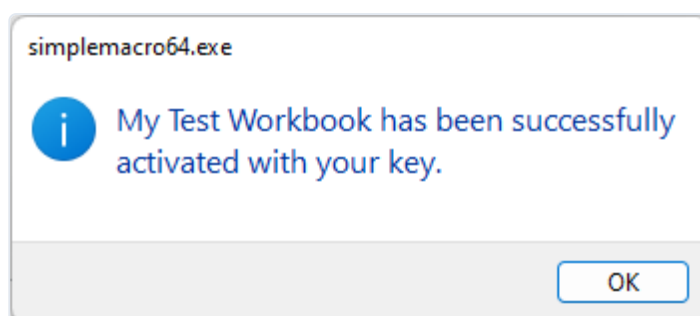


Si lo desea, puede [establecer restricciones en la clave](#), como limitar el número de ejecuciones o establecer una fecha de caducidad.

Haga clic en el botón **Generate** para crear una clave al instante. A continuación, puede copiar esta clave al portapapeles para enviarla a su cliente.



Cuando su cliente introduzca la clave y haga clic en **Activate**, se le concederá acceso al libro. La aplicación no volverá a solicitar la clave a menos que caduque.



Automatización de la generación de claves

Para facilitar la distribución de claves, hay disponible una aplicación de [generador de claves independiente](#) para los clientes registrados. También proporcionamos aplicaciones web gratuitas para generar claves automáticamente a través de su sitio web.

Claves de activación bloqueadas al hardware

Las **claves de activación bloqueadas por hardware** son una forma eficaz de evitar el uso compartido casual de su libro protegido, vinculando cada licencia a un equipo concreto. Esta guía le muestra cómo configurarlas.

Tutorial en vídeo disponible

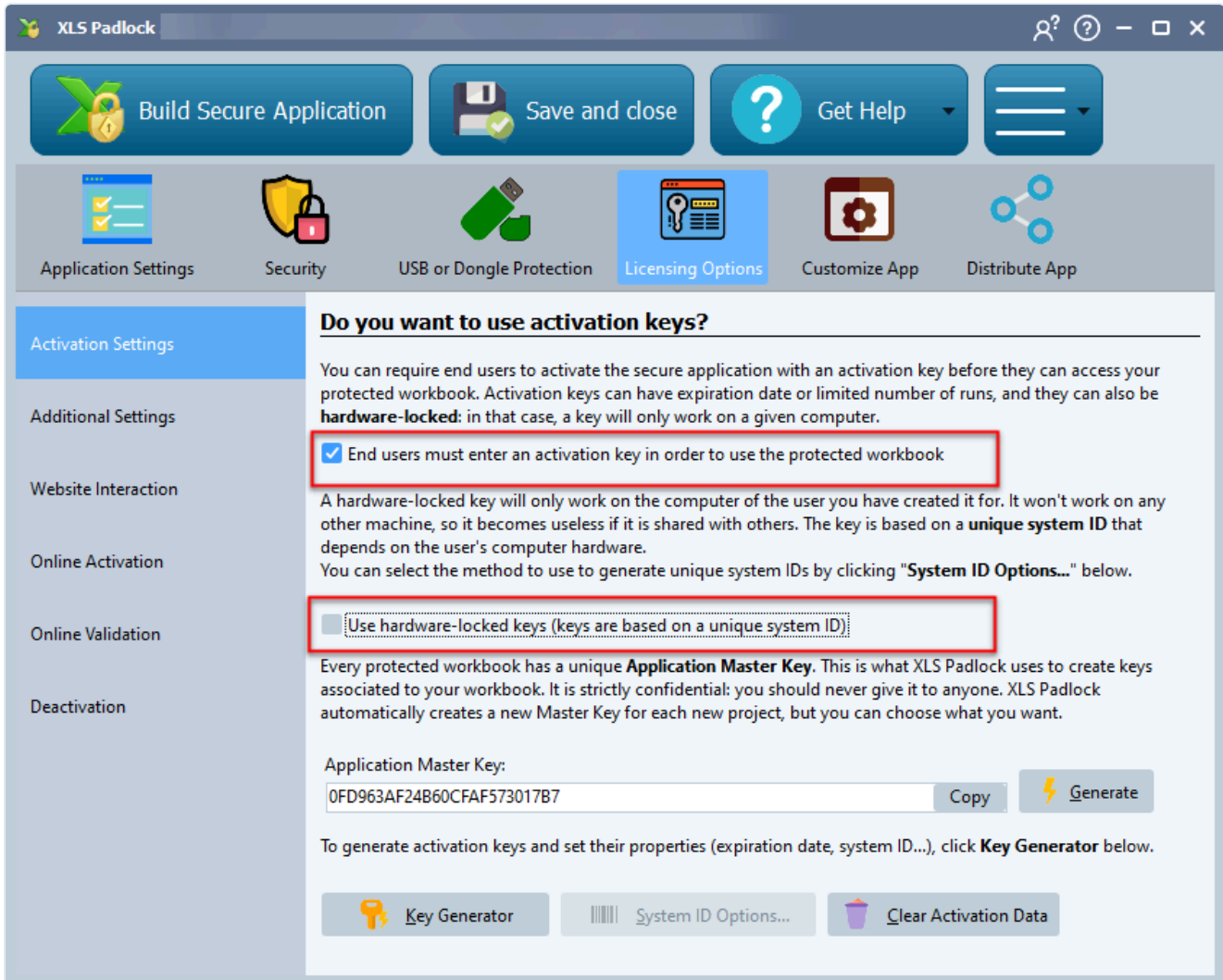
- [Vea nuestro tutorial en vídeo sobre el bloqueo por hardware de sus hojas de cálculo de Excel](#)
- [Vea una demostración en directo de esta función](#)

Acerca de las claves bloqueadas por hardware

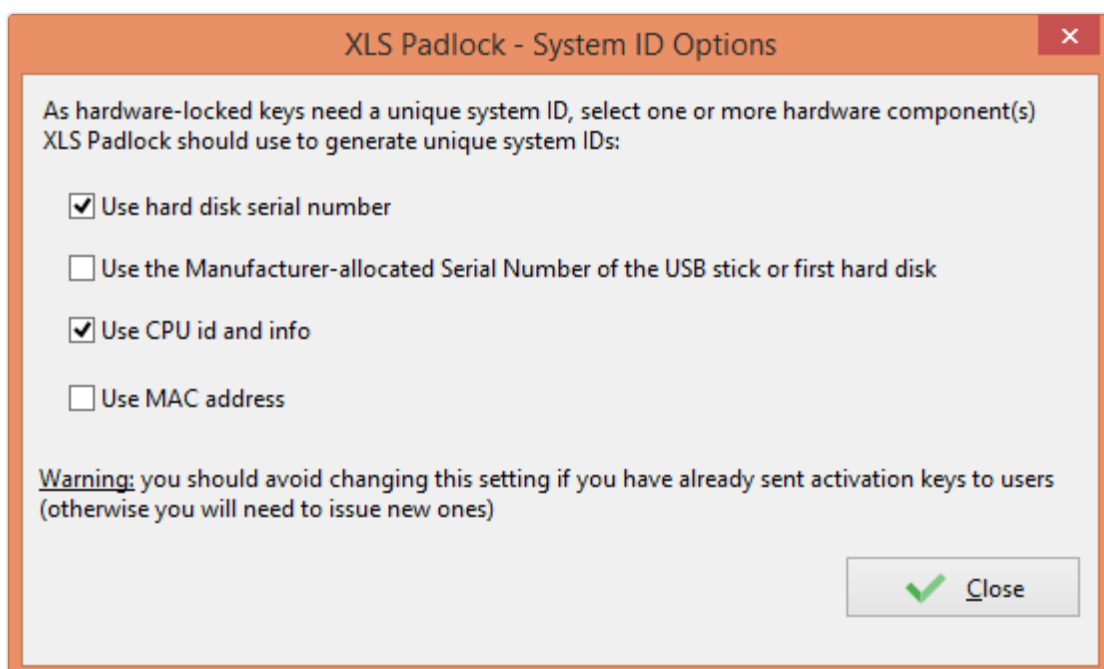
Una clave bloqueada por hardware **solo funcionará en el equipo para el que se generó**. Se basa en un **System ID** único derivado del hardware del equipo del usuario, lo que hace que la clave sea inútil si se comparte con otros.

1. Activar claves bloqueadas por hardware

En la configuración de XLS Padlock, marque tanto "End users must enter an activation key..." como "**Use hardware-locked keys**":



Opcionalmente, puede configurar qué componentes de hardware se utilizan para generar el System ID haciendo clic en "**System ID Options...**". Es importante no cambiar estas opciones después de haber comenzado a distribuir su aplicación.



Notas sobre los componentes del System ID

- **CPU:** Si la CPU no tiene un ID de serie integrado, se utiliza su información general. En hardware idéntico, esto podría dar como resultado el mismo ID. Es mejor combinar varias opciones de hardware.
- **MAC Address:** Si los usuarios disponen de varias formas de conectarse a internet (Wi-Fi, 4G, etc.), su dirección MAC puede cambiar, lo que invalidaría su clave.

Usar la huella de hardware mejorada

A partir de XLS Padlock 2026.0, el diálogo System ID Options también muestra una casilla adicional: **"Use enhanced hardware fingerprint"**.

Cuando esta opción está activada, el EXE protegido utiliza un cálculo reforzado del System ID:

- La fuente de bloqueo de Windows lee el identificador único de instalación de Windows directamente del registro del sistema en lugar del análisis del volumen de disco que utiliza el algoritmo heredado. Este identificador es estable frente a reformateos de volumen e intercambios físicos de disco: solo una reinstalación completa de Windows lo regenera.
- Todas las fuentes de hardware se normalizan (recortadas, con mayúsculas y minúsculas canonizadas) y se delimitan claramente dentro del cálculo de la huella, eliminando una clase de colisiones estructurales en las que dos máquinas distintas podían coincidir ocasionalmente en el mismo System ID. La nueva lógica también protege contra falsos negativos cuando una fuente queda temporalmente no disponible, por ejemplo cuando un antivirus bloquea las llamadas WMI, manteniendo el System ID estable en lugar de mutarlo.
- El hash subyacente se actualiza a SHA-256 en lugar del antiguo MD5, eliminando un algoritmo obsoleto de la huella criptográfica del EXE protegido.

El formato del System ID mostrado no cambia: sigue siendo de 14 caracteres hexadecimales en la conocida forma `XXXX-XXXX-XXXX`, por lo que sus usuarios finales y su flujo de generación de claves no notan ninguna diferencia.

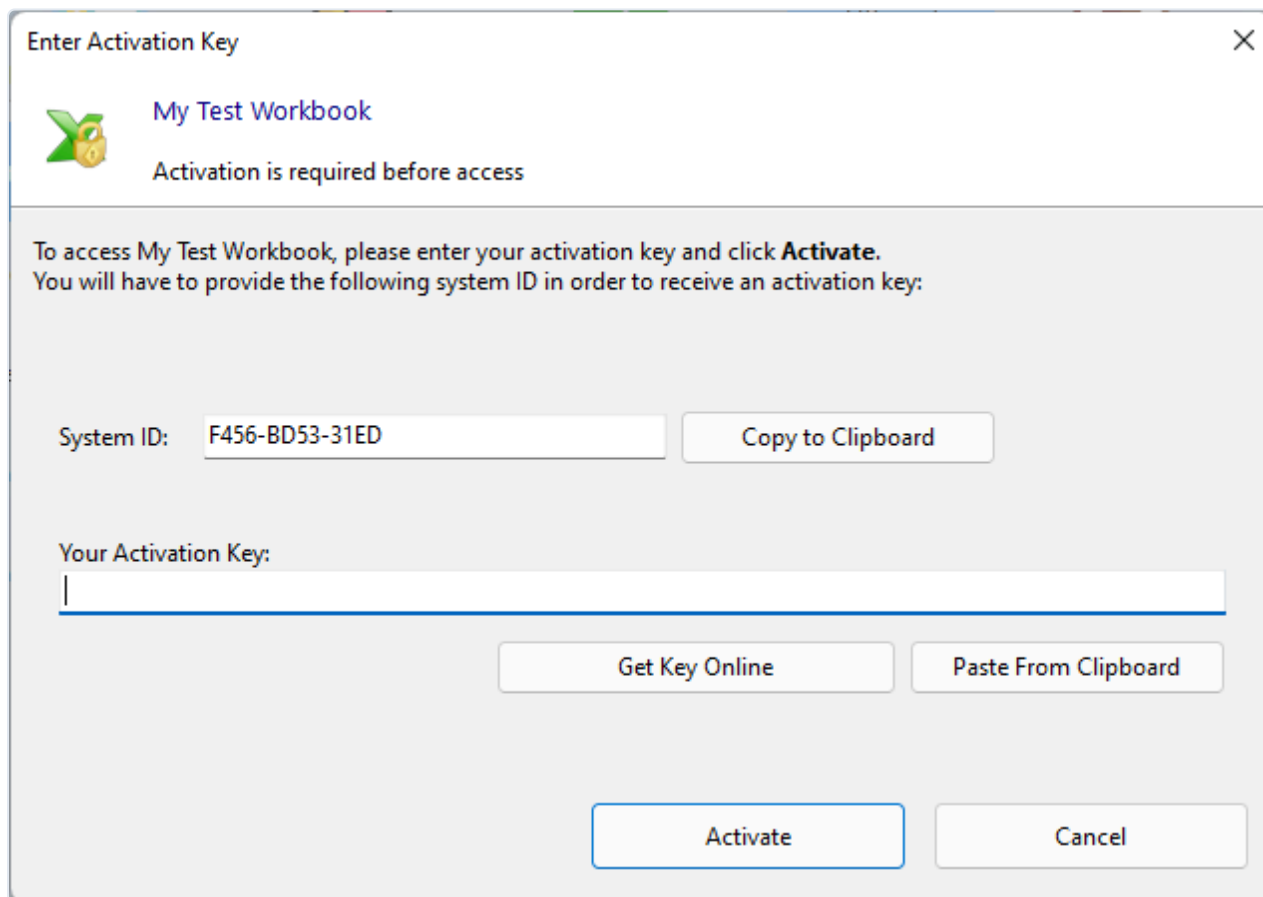
Comportamiento predeterminado:

- **Proyectos existentes** (guardados con XLS Padlock 2025.3 o anterior): la opción está **DESACTIVADA** de forma predeterminada. Esto conserva todas las claves de activación que ya están en circulación: los System ID que sus clientes obtuvieron de versiones anteriores siguen coincidiendo con las claves que emitió.
- **Proyectos nuevos** (creados con XLS Padlock 2026.0 y posteriores): la opción está **ACTIVADA** de forma predeterminada.

⚠ Importante: cambiar esta opción en un proyecto que ya se ha distribuido rotará el System ID de cada usuario final, invalidando todas las claves de activación que haya emitido hasta ahora. Actívela únicamente en un proyecto nuevo, o coordine el cambio con una reemisión de la clave de cada cliente.

2. Cómo obtienen los usuarios su System ID

Cuando un usuario ejecuta su aplicación por primera vez, se le pedirá que introduzca una clave de activación. Su System ID único se mostrará en esta ventana. El usuario debe copiar este ID y enviárselo a usted.



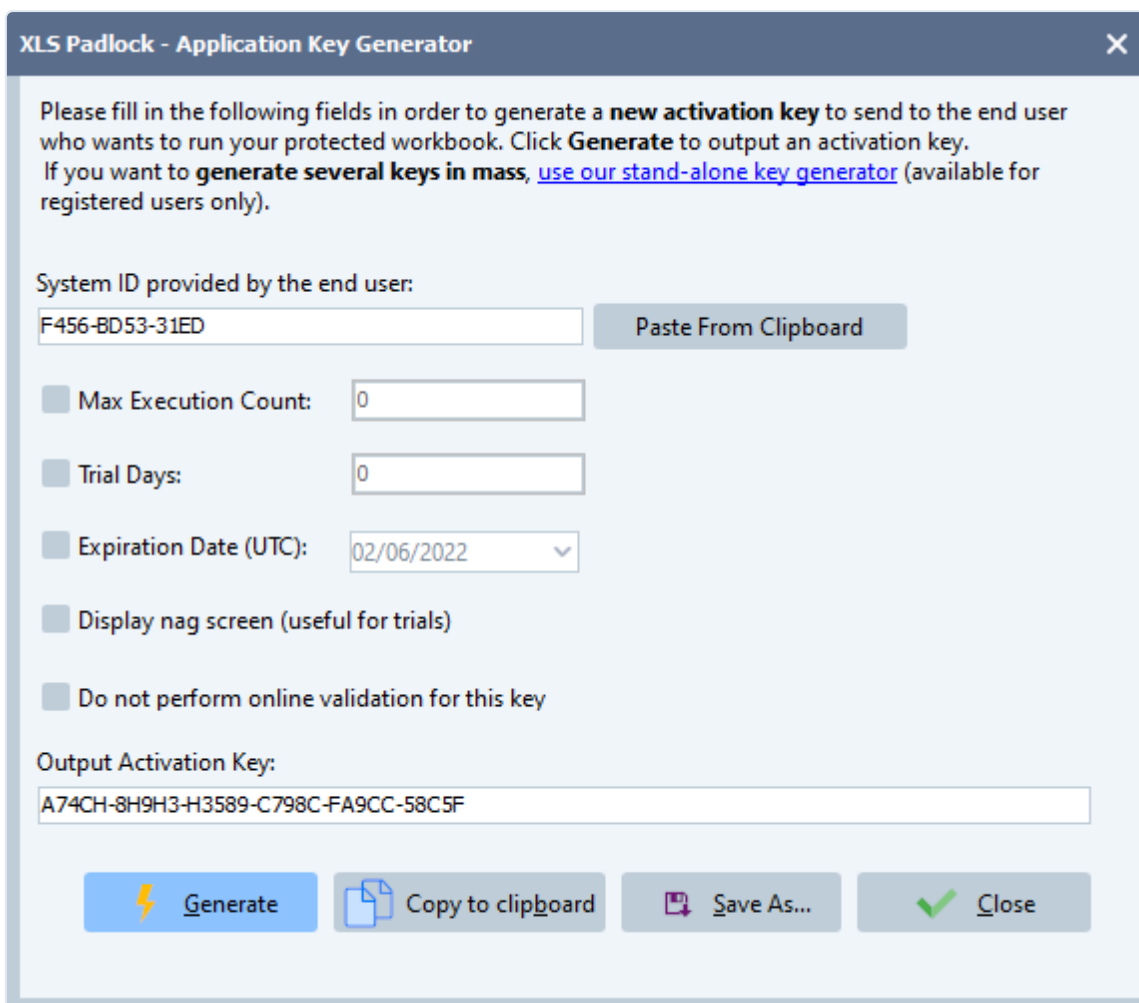
The screenshot shows a dialog box titled "Enter Activation Key" with a close button (X) in the top right corner. Below the title bar, there is a green Excel icon and the text "My Test Workbook" and "Activation is required before access". The main content area contains the following text: "To access My 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:". Below this text, there is a "System ID:" label followed by a text input field containing "F456-BD53-31ED" and a "Copy to Clipboard" button. Underneath, there is a "Your Activation Key:" label followed by a large empty text input field. At the bottom, there are four buttons: "Get Key Online", "Paste From Clipboard", "Activate", and "Cancel".

Automatizar la entrega de claves

Para evitar el intercambio manual de los System ID, puede configurar un botón "[Get Key Online](#)" que envíe al usuario a su sitio web con su System ID, o utilizar la función totalmente automatizada de [activación en línea](#).

3\ Generar la clave de activación

Abra el [Key Generator](#) en XLS Padlock. Pegue el System ID del cliente en el campo designado y haga clic en **Generate**. A continuación, puede copiar la clave y enviársela a su cliente.



Cuando el cliente introduce la clave, la aplicación se activará y no volverá a solicitar la clave (a menos que tenga una fecha de caducidad). Como la clave está vinculada a su System ID, no funcionará en ningún otro equipo.

¿Desea automatizar la creación de claves de activación bloqueadas por hardware? [Lea más sobre la activación en línea.](#)

Activación en línea

Cómo funciona

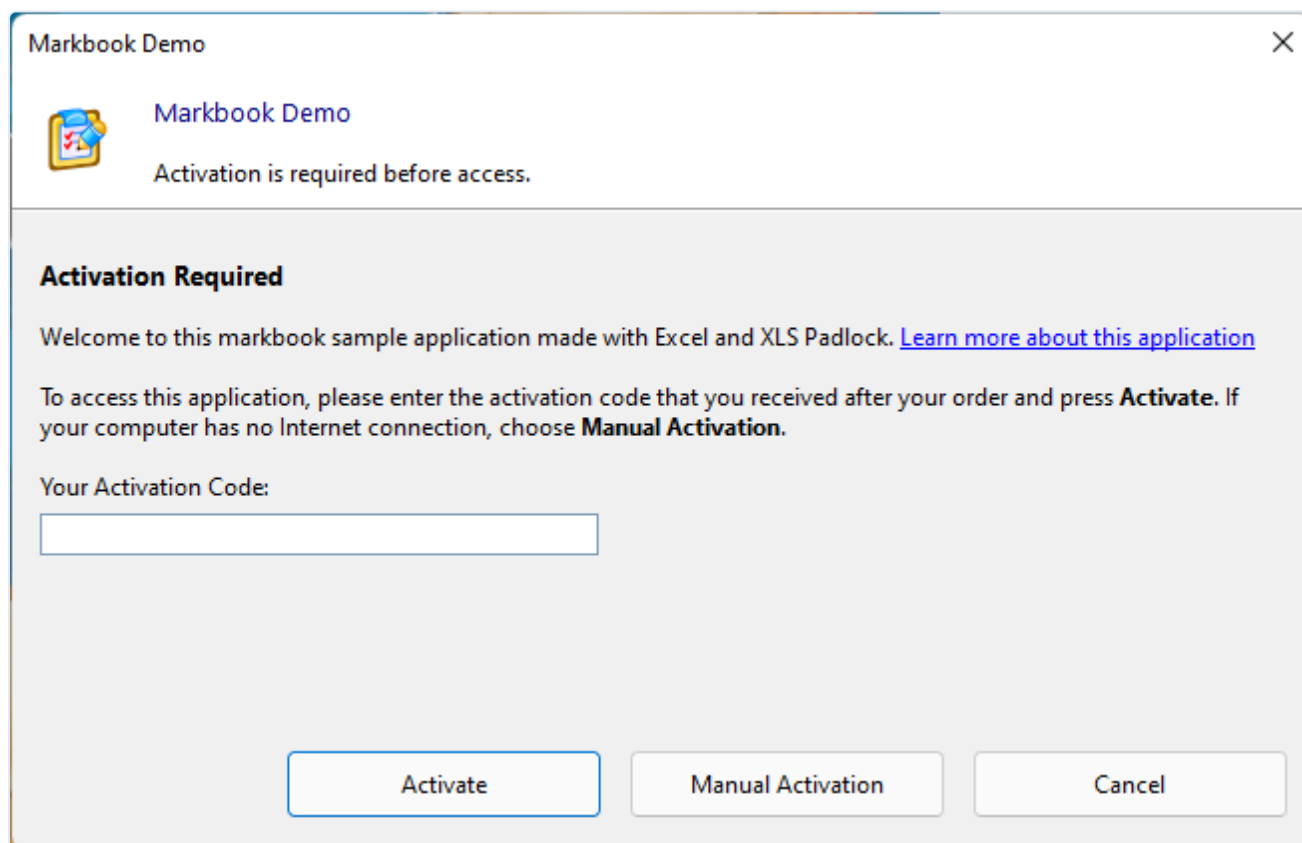
La activación en línea automatiza la obtención de [claves de activación](#) a través de internet. La aplicación protegida se comunica con su servidor web para descargar una clave de activación directamente, eliminando la necesidad de que los usuarios la introduzcan manualmente.

Se requiere un kit del lado del servidor

Para utilizar la activación en línea, debe instalar uno de nuestros kits del lado del servidor en su servidor web, como el XLS Padlock Activation Kit, el WooCommerce Integration Kit o el FastSpring Subscription kit. Puede descargar estos kits desde su [página de cuenta](#).

Si opta por el protocolo de activación 2026 (firma de respuestas Ed25519, véase más abajo), asegúrese de que su kit sea la versión 2026.0 o posterior. El kit detecta automáticamente el protocolo en cada solicitud, por lo que los libros heredados siguen funcionando con la misma instalación del kit.

👉 Cuando un usuario final inicia la aplicación, aparece un cuadro de diálogo que le solicita la activación. Este diálogo reemplaza al cuadro de diálogo estándar "Enter Activation Key" y es totalmente personalizable. Puede añadir sus propios campos para recopilar datos del usuario (como un número de pedido o una dirección de correo electrónico), que se enviarán a su servidor web. A continuación, su servidor valida esta información y, si tiene éxito, devuelve una clave de activación a la aplicación.



Markbook Demo

Markbook Demo
Activation is required before access.

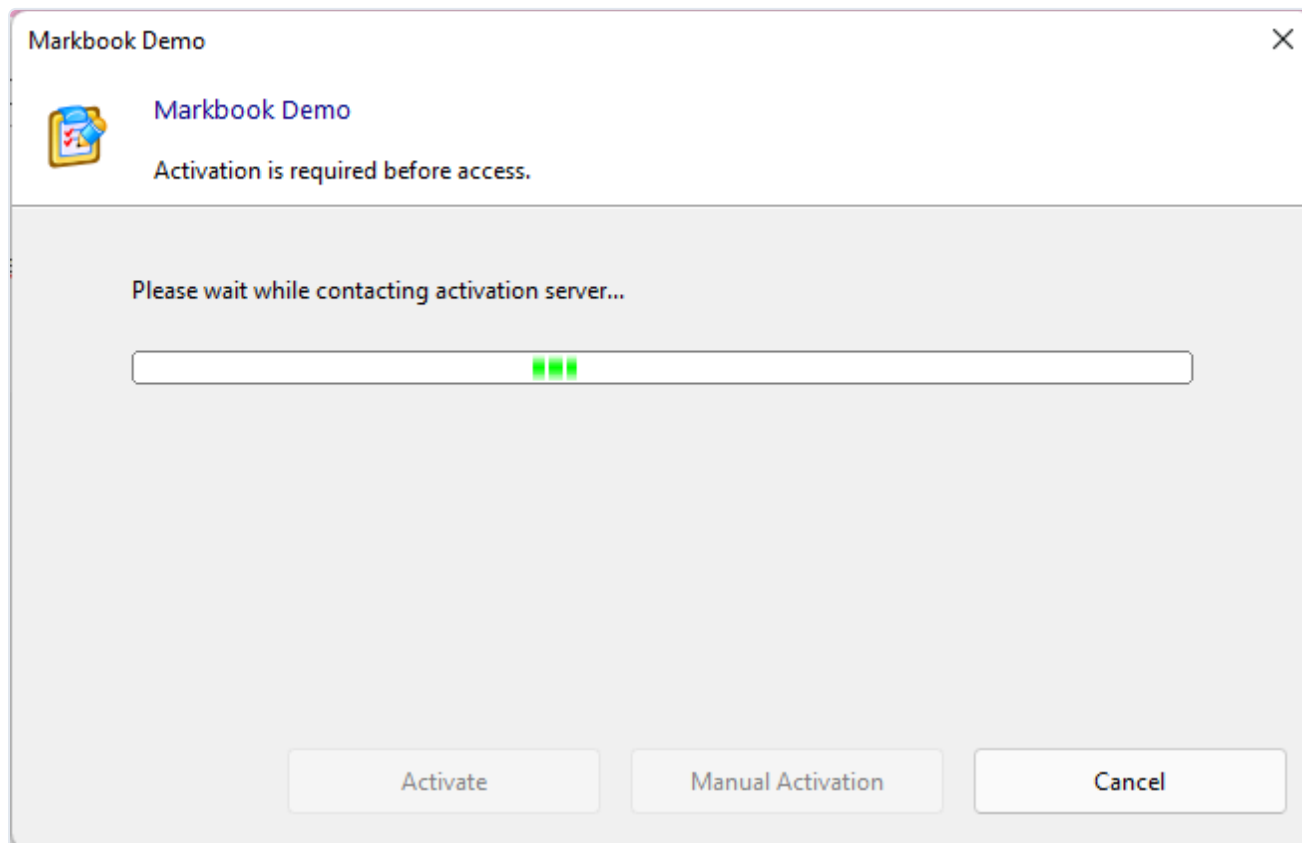
Activation Required

Welcome to this markbook sample application made with Excel and XLS Padlock. [Learn more about this application](#)

To access this application, please enter the activation code that you received after your order and press **Activate**. If your computer has no Internet connection, choose **Manual Activation**.

Your Activation Code:

Cuando el usuario hace clic en **Activate**, los datos se envían al servidor de activación:



Tras una activación correcta, se muestra un mensaje de confirmación y la aplicación se reinicia. Si se produce un error, aparecerá un cuadro de mensaje que permite al usuario volver a intentarlo.

Configuración

👉 Para activar la activación en línea, debe configurar las siguientes opciones:

Base Activation URL

Indique la URL completa del kit de activación instalado en su servidor. Por ejemplo, si instaló el kit en una subcarpeta denominada "activation", la URL sería

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

Use HTTPS

Se admiten conexiones seguras mediante TLS/SSL. Siempre debe utilizar URL que comiencen por `https://` .

⚠️ Deje el campo en blanco si ****no**** desea utilizar la activación en línea.

Identificación del cliente: Security Key o keypair Ed25519

A partir de XLS Padlock 2026.0, puede elegir entre dos regímenes de protocolo de activación mediante la opción **Compatibility mode for pre-2026 activation kits**:

- **Compatibility mode marcado** (predeterminado, recomendado si su servidor de activación es anterior a la versión 2026): el campo GUID heredado **Security Private Key** identifica su aplicación ante el kit de activación. El protocolo está basado en líneas con respuestas sin firmar. Este es el comportamiento de todas las versiones de XLS Padlock anteriores a la 2026.0.
- **Compatibility mode sin marcar** (requiere la versión 2026 del XLS Padlock Activation Kit, WooCommerce Integration Kit o FastSpring Subscription Kit): el campo heredado se oculta y un **keypair Ed25519** por proyecto toma el control. Haga clic en el botón **Generate keypair** en la página Online Activation para producir un keypair nuevo. La clave pública se incrusta en el libro protegido; la clave privada se le muestra en un diálogo de una sola vez para que pueda pegarla en la configuración de su kit de activación (`xlspadlocksignkey` en `config.ini`).

Si pierde la clave privada, haga clic en **Show config.ini snippet** en la misma página para volver a mostrarla. La clave se conserva en su archivo de proyecto `.xplp`. Trate el `.xplp` como información confidencial: no lo incluya en el control de versiones, no lo envíe por correo electrónico para soporte.

El protocolo 2026 utiliza un sobre de solicitud JSON y verifica una firma Ed25519 separada en cada respuesta del servidor de activación. Esto protege a sus clientes contra respuestas falsas de "activado" o mensajes de error de phishing que un atacante a nivel de red podría inyectar de otro modo eludiendo TLS (proxy corporativo con inyección de CA, antivirus malicioso, certificado raíz comprometido).

Allow Manual Activation if No Internet Connection

Es posible que algunos usuarios no dispongan de una conexión a internet activa. Para permitirles activar manualmente, habilite la opción "**Allow Manual Activation if No Internet Connection**".

Este método manual funciona igual que las claves de registro estándar. En este caso, debe estar preparado para gestionar las solicitudes de activación de los usuarios que estén sin conexión.

Advertencia

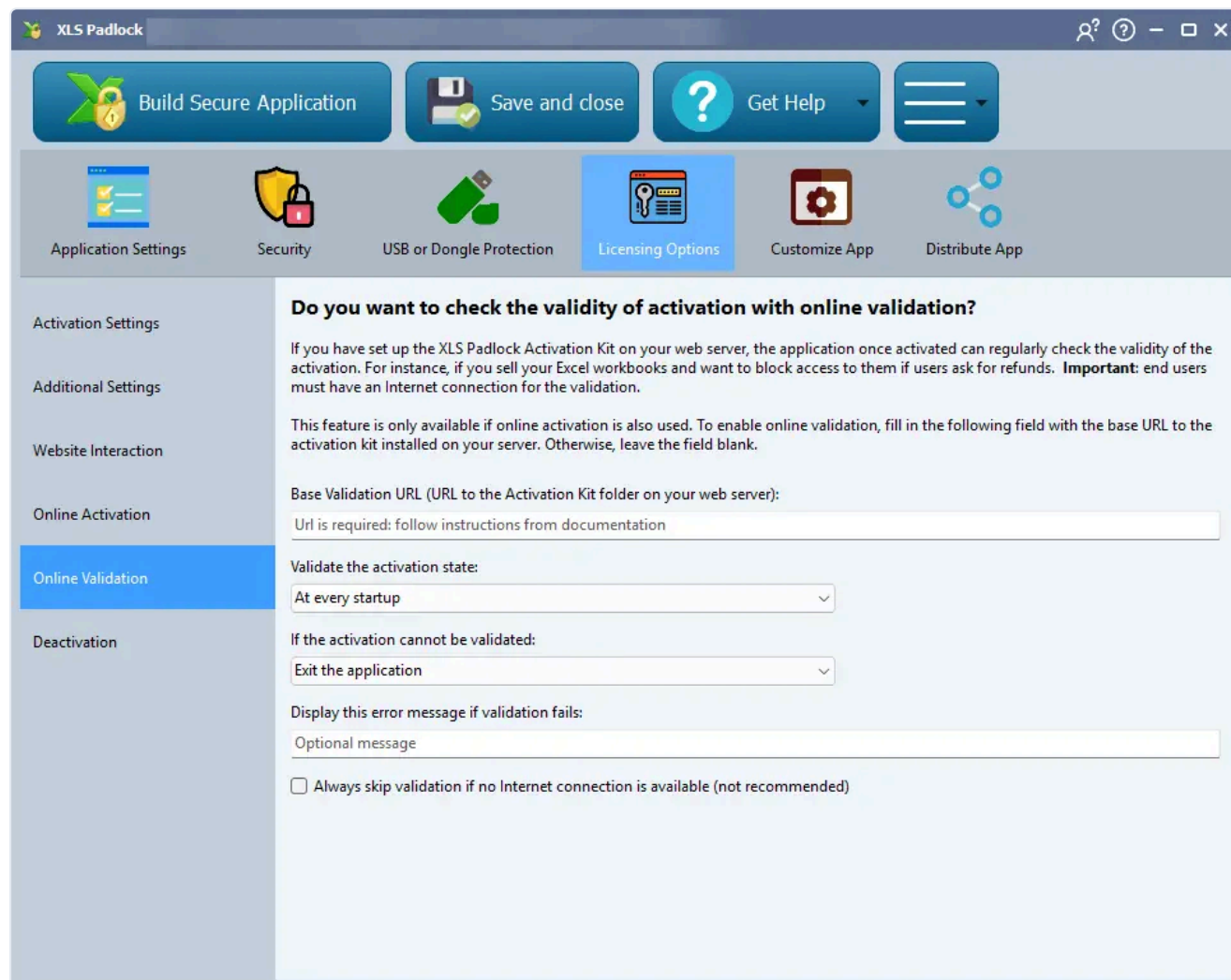
Si acepta activaciones manuales, asegúrese de generar claves de activación que **no realicen validación en línea** (opción en el [generador de claves](#)). De lo contrario, la validación fallará.

Véase también

- [Cómo vender libros de Excel con suscripciones](#)
- [Mostrar un botón "Purchase Online" en la pantalla de aviso](#)

Validación en línea

La validación en línea proporciona **control remoto** sobre sus aplicaciones activadas. Cuando se utiliza con uno de nuestros kits de activación del lado del servidor (como el XLS Padlock Activation Kit, el WooCommerce Kit o el [FastSpring subscription kit](#)), su aplicación puede comprobar periódicamente con su servidor que su activación sigue siendo válida.



Advertencia

Esta función requiere que la [activación en línea](#) esté habilitada y configurada previamente.

Esto resulta útil en varios escenarios:

- Puede desactivar de forma remota una aplicación si un cliente solicita un reembolso.
- Puede aplicar un acceso basado en suscripción, en el que la aplicación deja de funcionar si una suscripción caduca.

Cómo funciona

- La validación en línea utiliza el mismo token único que se generó durante la activación en línea inicial para identificar el dispositivo del cliente.
- La clave de activación en sí nunca se envía a través de internet durante la validación.
- Se requiere una conexión a internet activa para que se produzca la comprobación de validación.

Configuración

👉 Debe configurar la **Base Validation URL** para que apunte a su script de validación del lado del servidor. Normalmente es la URL del XLS Padlock Activation Kit, el XLS Padlock WooCommerce Integration Kit o el FastSpring subscription kit en su servidor web. Por ejemplo, si instaló el kit de activación en una subcarpeta denominada "activation", la URL sería

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

Use HTTPS

Se admiten conexiones seguras mediante TLS/SSL. Siempre debe utilizar URL que comiencen por `https://` .

⚠ Deje el campo en blanco si ****no**** desea utilizar la validación en línea.

Frecuencia de validación

Elija cuándo debe realizar la aplicación una validación: en cada inicio, de forma aleatoria, cada X días o cada X ejecuciones. Debe especificar el valor de X donde sea necesario.

Advertencia

Tras la activación, se requiere una validación inicial en el siguiente inicio. Después de esta primera comprobación, la aplicación se ajustará a la frecuencia de validación seleccionada.

Acción en caso de fallo de validación

Defina lo que debe hacer la aplicación si falla la validación en línea: * **Exit the application:** La aplicación se cerrará inmediatamente. * **Blacklist activation key:** La clave actual quedará invalidada y se pedirá al usuario que introduzca una nueva. Si el usuario vuelve a introducir la misma clave, se intentará la validación de nuevo. * **Do nothing:** La aplicación seguirá ejecutándose. Puede utilizar las extensiones VBA de XLS Padlock para comprobar el estado de validación e implementar una lógica personalizada.

Si la activación no se puede validar

Puede mostrar un mensaje de error personalizado al usuario si la validación falla, indicándole qué debe hacer a continuación.

Omitir la validación si está sin conexión

Puede permitir que la aplicación omita la comprobación de validación si no se detecta ninguna conexión a internet. Por lo general, esto no se recomienda por motivos de seguridad.

Advertencia

Si esta opción está deshabilitada, los usuarios deben disponer de una conexión a internet activa para el proceso de validación. De lo contrario, la validación fallará.

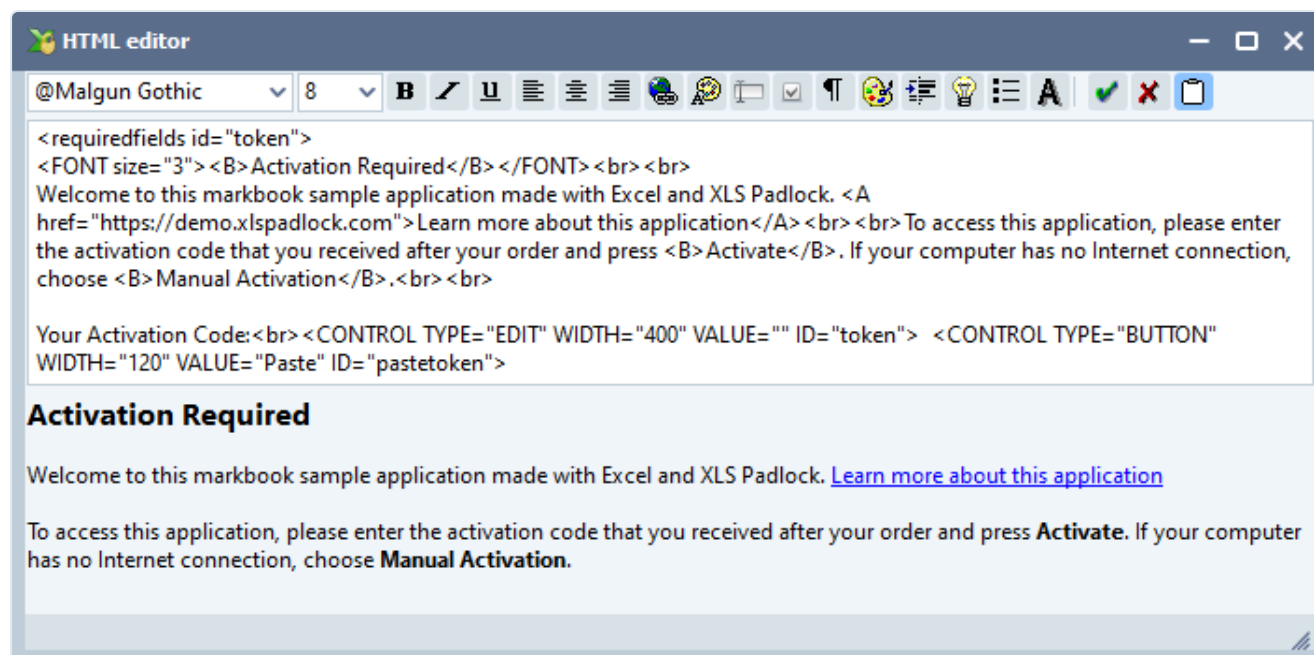
VBA: Recuperar información de suscripción/licencia

Tras una validación en línea correcta, puede recuperar datos personalizados enviados desde su servidor mediante la llamada VBA `XLSPadlock.PLEvalVar("ValidationAddServerData")`. Esto resulta útil para transmitir información específica de suscripción al libro.

Editor del formulario de registro

El Editor del formulario de registro le permite personalizar el texto que aparece en el [cuadro de diálogo de activación en línea](#). El editor admite HTML básico, por lo que puede usar etiquetas como `` , `
` , `` , etc.

XLS Padlock proporciona un editor HTML básico con una vista previa en vivo del texto del cuadro de diálogo:



Controles personalizados

Puede añadir campos personalizados para solicitar información adicional a los usuarios, que se enviará a su servidor web. También puede añadir un botón para pegar texto en el campo del token.

- **Botón Paste:** `<CONTROL TYPE="BUTTON" WIDTH="120" VALUE="Paste" ID="pastetoken">`
- **Campo de texto:** `<CONTROL TYPE="EDIT" WIDTH="400" VALUE="" ID="token">`

Campos obligatorios

Puede hacer que los campos sean obligatorios indicando sus ID en una etiqueta `<requiredfields>` . Por ejemplo, para exigir los campos "token" y "name":

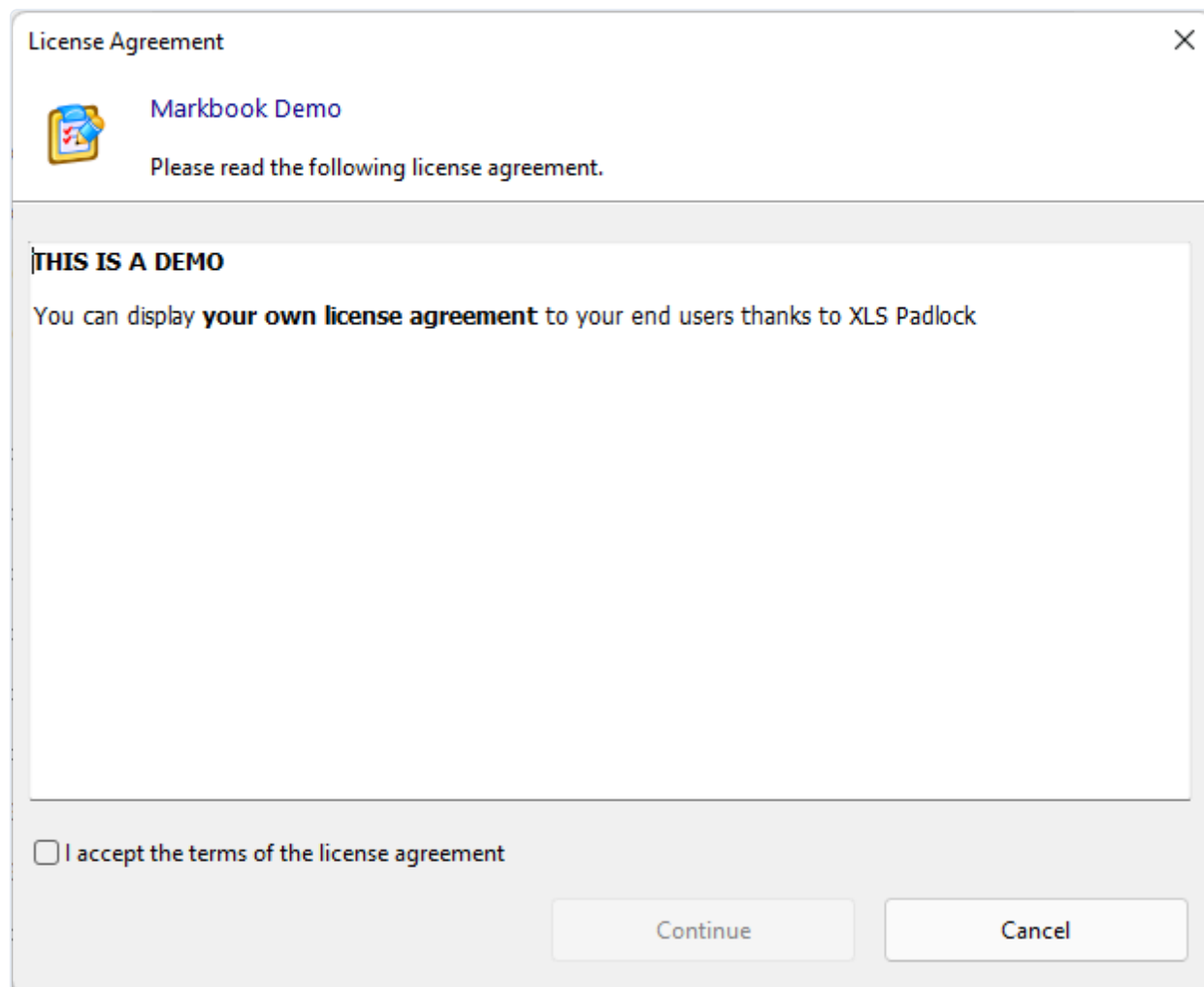
```
<requiredfields id="token,name">
```

Ejemplo completo

Mostrar el acuerdo de licencia

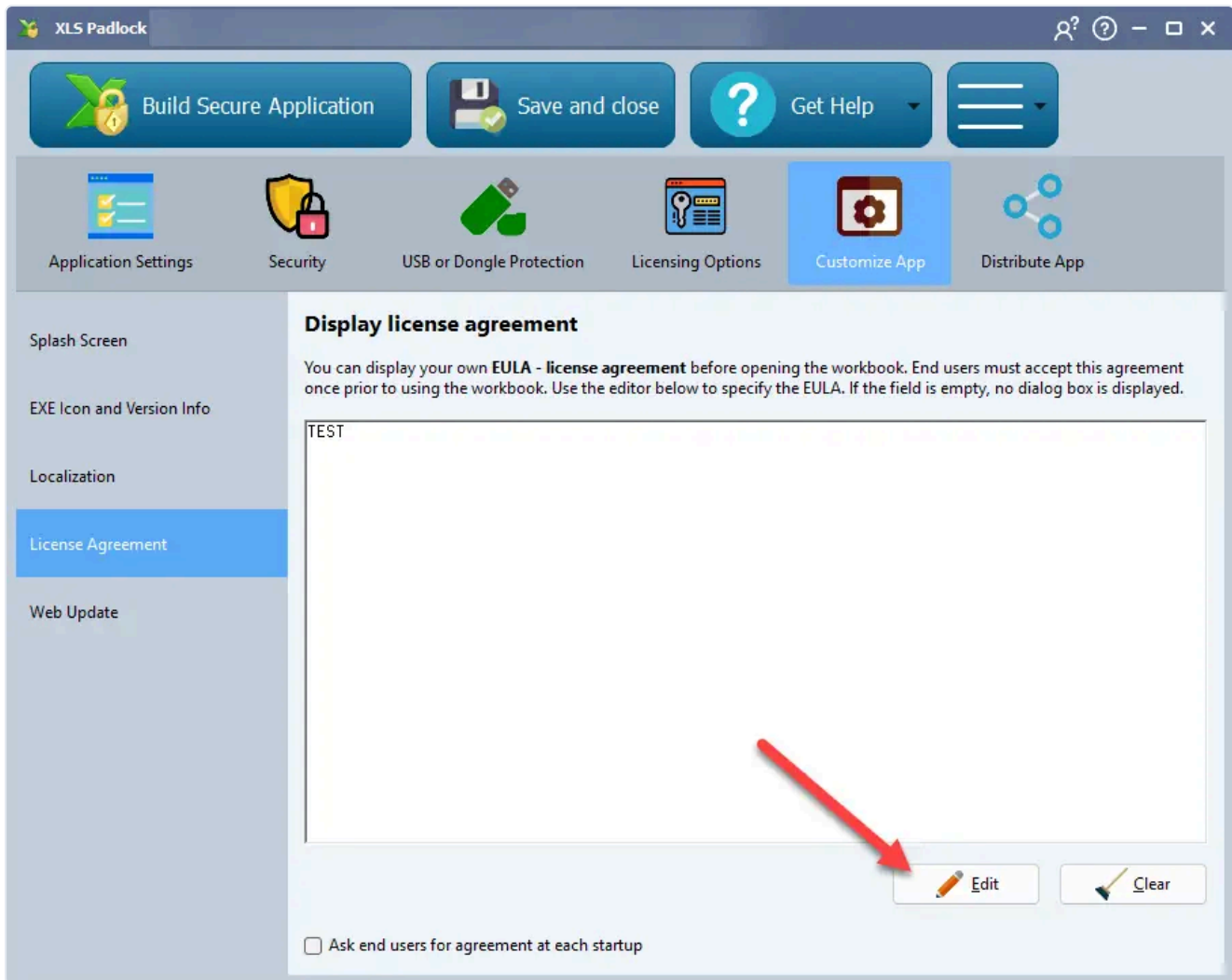
Puede exigir a los usuarios que acepten un **acuerdo de licencia de usuario final (EULA)** antes de poder abrir su libro protegido.

El libro compilado mostrará un cuadro de diálogo como el que se muestra a continuación:



Cuando el usuario marca "I accept the terms of the license agreement", el botón "Continue" se activa, lo que le permite continuar.

👉 Para añadir su EULA, haga clic en **Edit** para abrir un editor de texto enriquecido en esta página:



Puede cargar contenido desde un archivo RTF o pegar texto directamente desde una aplicación como Microsoft Word.

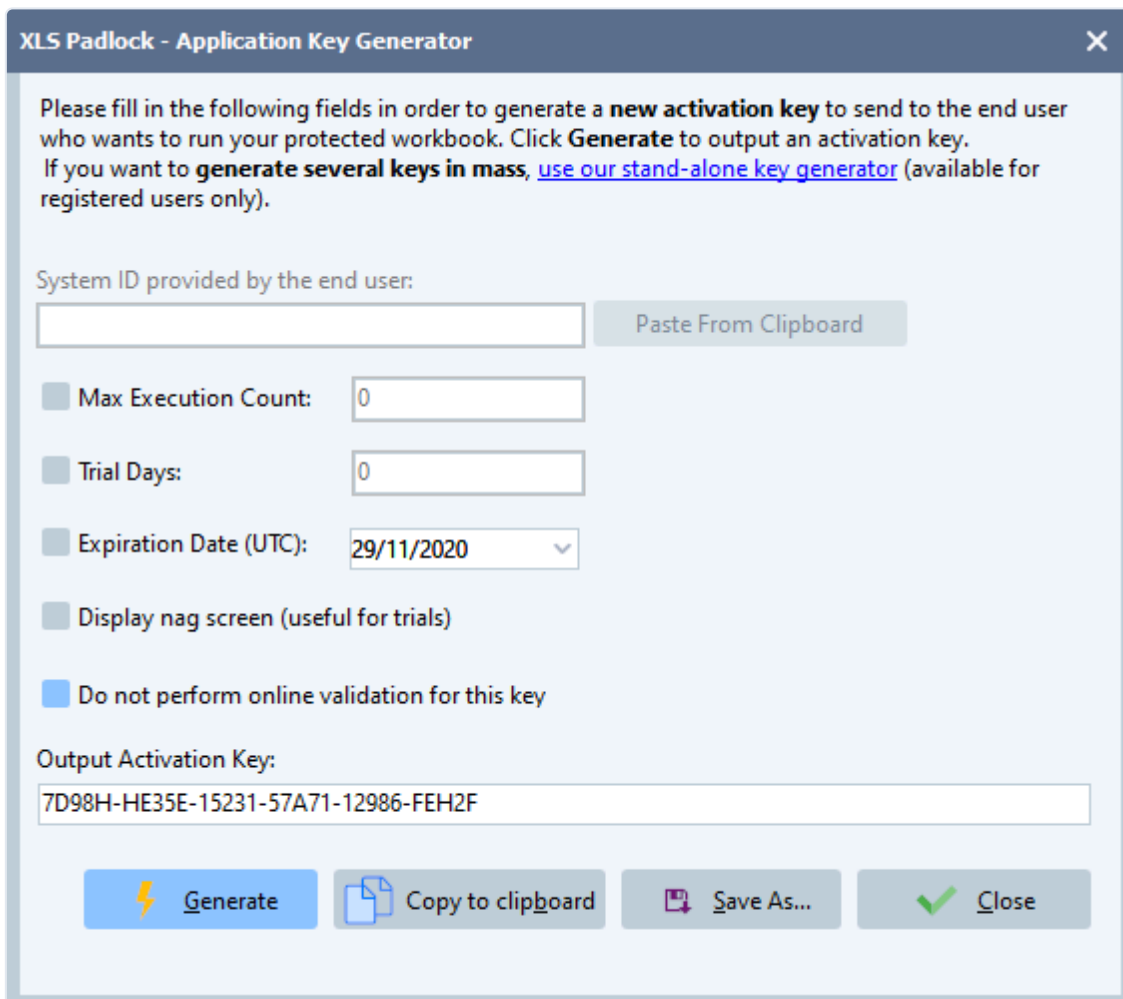
NOTA

Si el contenido del EULA se deja vacío, no se mostrará ningún cuadro de diálogo.

i Para mostrar el EULA cada vez que se inicia la aplicación, active la opción **Ask end users for agreement at each startup**. De forma predeterminada, los usuarios solo necesitan aceptarlo una vez.

Generador de claves (portátil y servidor)

XLS Padlock incluye un **Key Generator** integrado para crear claves de activación para sus aplicaciones de libro.



- **Para claves de activación simples**, basta con hacer clic en el botón **Generate** para crear una clave al instante.
- **Para claves vinculadas al hardware**, introduzca el System ID proporcionado por su usuario y, a continuación, haga clic en **Generate**. La clave resultante quedará vinculada a ese ordenador concreto.

Puede copiar la clave generada al portapapeles para enviarla a su cliente, o guardarla como archivo `.txt` para sus registros.

Las claves son específicas de cada libro

El generador de claves crea claves de activación solo para el proyecto de libro abierto actualmente. Las claves generadas para un libro no funcionarán con otro a menos que compartan la misma [Application Master Key](#).

SDK del generador de claves

Ofrecemos un SDK del Generador de Claves gratuito (una aplicación web basada en PHP) que puede instalar en su propio sitio web. Este SDK le permite automatizar la generación y la entrega de claves de activación para sus libros.

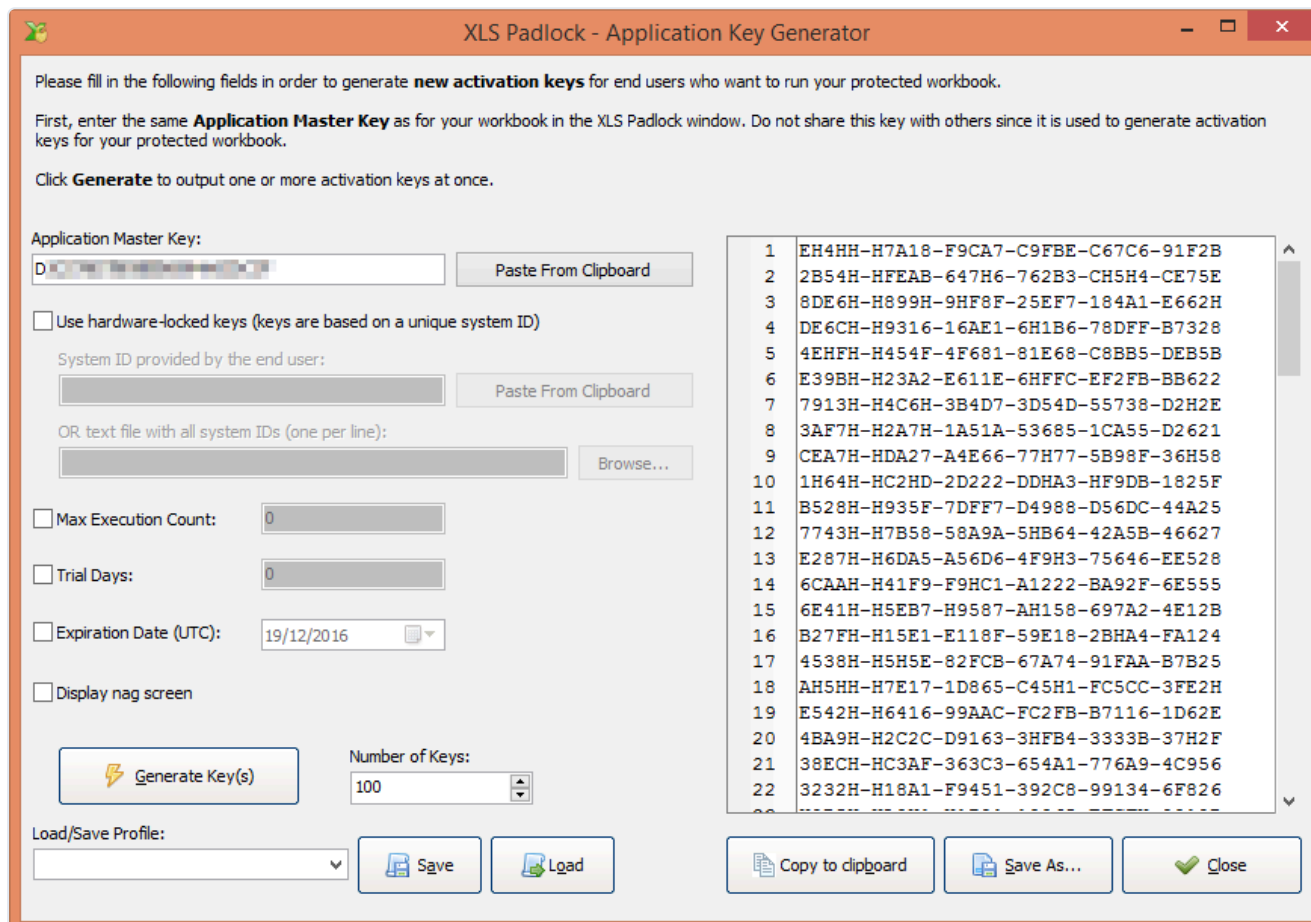
A diferencia del [Generador de Claves integrado](#), que crea claves manualmente dentro de XLS Padlock, este SDK se ejecuta en su propio servidor para emitir claves de forma automática.

Para una solución de licencias totalmente automatizada, considere combinar el SDK con la [función de Activación en línea](#).

👉 Puede descargar el SDK del Generador de Claves desde su [página de cuenta](#).

Generador de claves autónomo

XLS Padlock ofrece un generador de claves independiente para los clientes registrados. Esta herramienta le permite [crear claves de activación](#) para sus libros protegidos sin tener que abrir Excel ni el complemento de XLS Padlock.



Para una gestión más sencilla, el generador de claves independiente también le permite guardar la configuración como **profiles**.

👉 Puede descargar el generador de claves independiente desde su [página de cuenta](#).

Restricciones de las claves

Cuando genera [claves de activación](#) con XLS Padlock, puede **aplicar varias restricciones** mediante las siguientes opciones:

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:

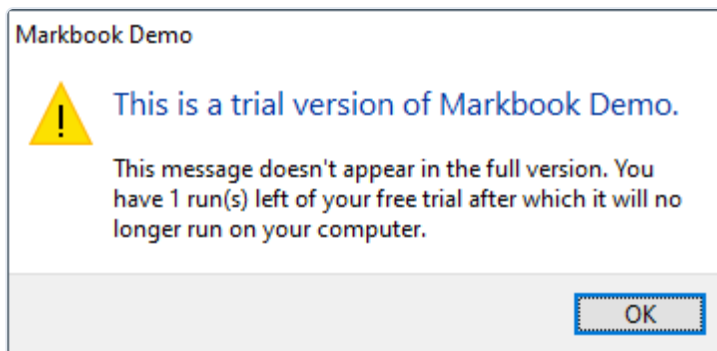
Expiration Date (UTC):

Display nag screen (useful for trials)

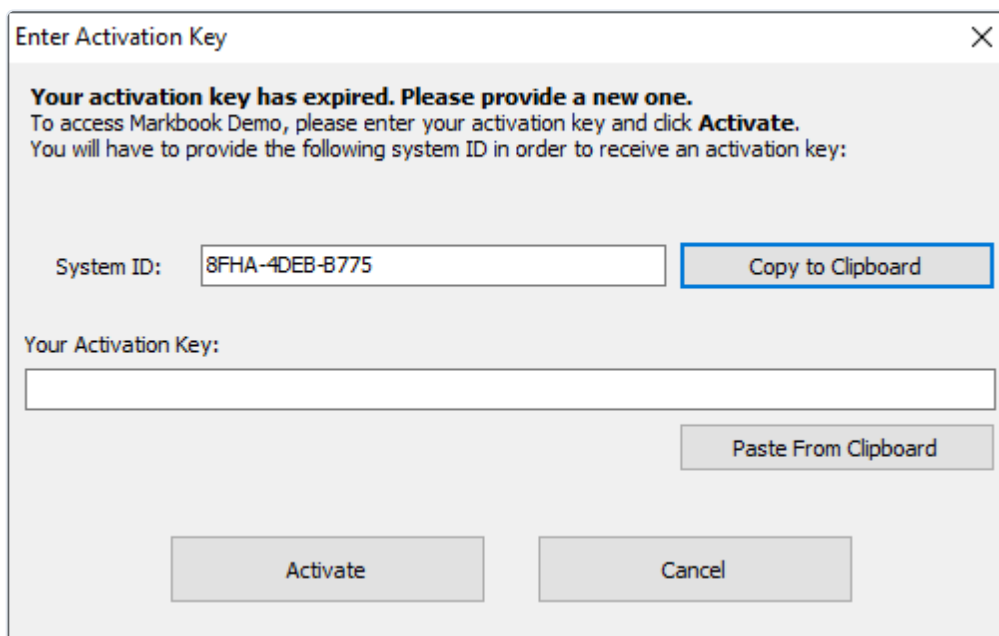
Do not perform online validation for this key

Output Activation Key:

- 👉 Limite cuántas veces se puede ejecutar su aplicación activando **"Max Execution Count"** e introduciendo el número deseado.
- 👉 Establezca que la clave caduque después de un número específico de días (active **"Trial Days"**) o en una fecha concreta (active **"Expiration Date"**).
- 👉 Muestre un cuadro de diálogo recordatorio (o "nag screen") al inicio activando **"Display Nag Screen"**. Esto es [útil para las versiones de prueba](#), ya que informa al usuario de los días o las ejecuciones restantes.



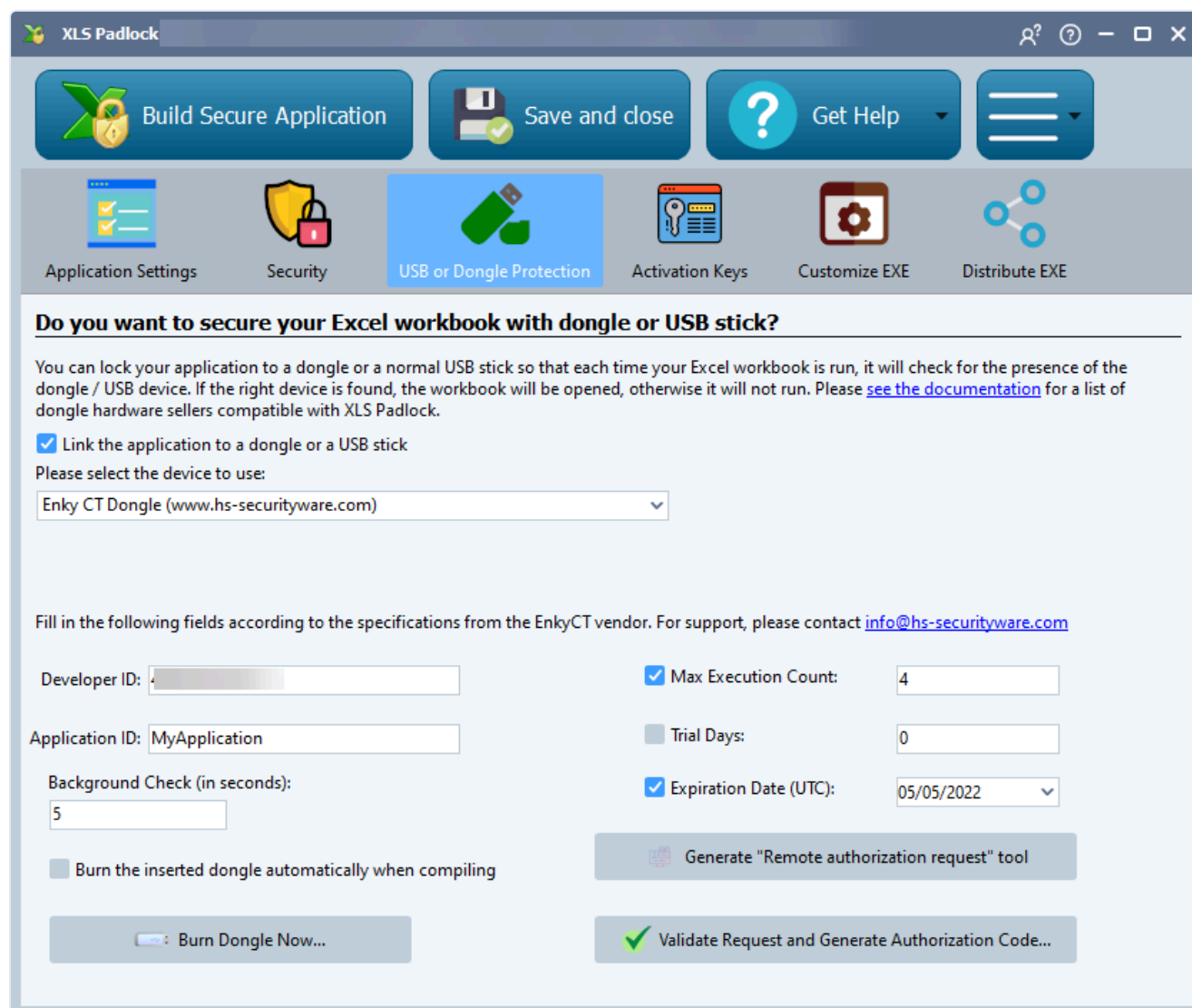
Por último, si utiliza la [validación en línea](#) y necesita eximir una clave concreta de esta comprobación (por ejemplo, para un usuario en un entorno sin conexión), puede activar "**Do not perform online validation for this key**". Esto solo se recomienda cuando se usa en combinación con claves vinculadas al hardware. Cuando se alcanza el límite de una clave (fecha, número de ejecuciones o días de prueba), se solicitará al usuario que introduzca una nueva clave para seguir usando la aplicación:



Protección por USB o mochila (dongle)

Con XLS Padlock, puede vincular sus libros de Excel a una memoria USB o a una mochila de seguridad dedicada. Esto significa que **la aplicación del libro compilado no funcionará si no se ha insertado el dispositivo USB correcto**. La aplicación comprobará la presencia del dispositivo al iniciarse y también puede comprobarla de forma periódica durante la ejecución.

Para utilizar esta protección, marque la opción "**Link the application to a dongle or a USB stick**" (Vincular la aplicación a una mochila o memoria USB) en la página "USB Protection" y seleccione el modelo de dispositivo que tiene.



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 main content area is titled 'Do you want to secure your Excel workbook with dongle or USB stick?'. It contains the following text: 'You can lock your application to a dongle or a normal USB stick so that each time your Excel workbook is run, it will check for the presence of the dongle / USB device. If the right device is found, the workbook will be opened, otherwise it will not run. Please [see the documentation](#) for a list of dongle hardware sellers compatible with XLS Padlock.' Below this is a checked checkbox 'Link the application to a dongle or a USB stick' and a dropdown menu 'Please select the device to use:' with 'Enky CT Dongle (www.hs-securityware.com)' selected. Further down, there is a note: 'Fill in the following fields according to the specifications from the EnkyCT vendor. For support, please contact info@hs-securityware.com'. The form includes several fields: 'Developer ID:' (empty), 'Application ID:' (MyApplication), 'Background Check (in seconds):' (5), 'Max Execution Count:' (checked, 4), 'Trial Days:' (unchecked, 0), and 'Expiration Date (UTC):' (checked, 05/05/2022). At the bottom, there are two buttons: 'Burn Dongle Now...' and 'Generate "Remote authorization request" tool'. A 'Validate Request and Generate Authorization Code...' button is also visible at the bottom right.

XLS Padlock admite varios tipos de protección basada en hardware para vincular su aplicación de Excel a un dispositivo físico.

Mochila Enky CT

Según el proveedor, Enky CT es una mochila de reloj basada en tarjeta inteligente de 32 bits, fácil de usar y flexible, utilizada principalmente para la protección de software y la limitación temporal.

Para obtener asistencia, póngase en contacto con info@hs-securityware.com o visite <https://hs-securityware.com>.

Cómo usar la mochila Enky CT

Introduzca el **Developer ID** proporcionado por HS Securityware (distingue entre mayúsculas y minúsculas).

El **Application ID** puede ser cualquier valor que desee para identificar la mochila, y debería estar relacionado con la aplicación que está creando.

Background Check

La aplicación comprueba la presencia de la mochila al iniciarse y también puede comprobarla de forma periódica durante la ejecución. Puede definir la frecuencia de estas comprobaciones; por ejemplo, un valor de 20 significa que la aplicación comprobará la mochila cada 20 segundos. Si no se encuentra la mochila, aparece una pantalla recordatoria que exige al usuario que inserte la mochila en un plazo de 15 segundos, o la aplicación se cerrará.

Set restrictions on the application

Puede limitar el número de veces que se puede ejecutar su aplicación estableciendo un **Max Execution Count**. También puede hacer que su aplicación caduque tras un número determinado de días (**Trial Days**) o en una **Expiration Date** concreta.

Funciones de actualización remota de mochilas

Este modelo de mochila le permite actualizar de forma remota las mochilas que ya están en poder de sus clientes. Por ejemplo, puede ampliar una fecha de caducidad o añadir más ejecuciones. Esta función de actualización remota se basa en un sistema de código de solicitud y autorización.

Cómo actualizar de forma remota la mochila de un cliente:

1. En primer lugar, genere la "remote authorization request tool" (herramienta de solicitud de autorización remota) desde XLS Padlock y envíe a su cliente los archivos EXE y DAT resultantes.

Aviso

Estos dos archivos (`.EXE` y `.DAT`) deben permanecer en la misma carpeta para que la herramienta funcione.

2. El cliente ejecuta la herramienta para generar un **request code** (código de solicitud), que le envía a usted.
3. En XLS Padlock, elija "Validate Request and Generate Authorization Code", pegue el código de solicitud y haga clic en "Validate Request Code". A continuación, podrá ver el estado actual de la mochila y establecer nuevos límites (número de usos o una nueva fecha de caducidad).
4. Genere el **authorization code** (código de autorización) y envíelo de vuelta al cliente. Este utilizará la misma herramienta para introducir el código y aplicar la actualización a su mochila.

Mochila Enky LC

Para obtener asistencia, póngase en contacto con info@hs-securityware.com o visite <https://hs-securityware.com>.

La Enky LC es una mochila HID económica y sin controladores para la protección de software.

Cómo usar la mochila Enky LC2

1. Introduzca el **Developer ID** que recibió de HS-Security Ware.
2. Introduzca un **Product ID** único para su libro. Esto garantiza que solo se aceptarán las mochilas con el Product ID correcto.

Pasos para configurar una mochila Enky LC2

Para vincular una mochila a su aplicación, primero debe "grabarla" mediante el botón "**Burn Dongle Now**" de XLS Padlock. Esto configura la mochila con sus identificadores. Este proceso también puede producirse automáticamente al compilar su aplicación si hay conectada una mochila compatible.

Background Check

Puede activar una comprobación en segundo plano para garantizar que la mochila permanezca conectada mientras la aplicación se está ejecutando. Puede establecer el intervalo (en segundos) con el que se realiza la comprobación.

Protección con memoria USB genérica

Puede vincular su aplicación a una o varias memorias USB genéricas en función de su identificador único de fabricante.

- En primer lugar, debe introducir un **Application Secret ID**. Este se combina con el identificador de la memoria USB para generar un hash único.
- Cuando se inicia la aplicación, esta examina todas las unidades USB. Si se encuentra una unidad con un hash autorizado, el libro se abre.

Para autorizar una memoria USB, conéctela, selecciónela en la lista de unidades de XLS Padlock y haga clic en "Allow this USB disk". Puede autorizar varias memorias USB.

NOTA

La protección con memoria USB ofrece menos flexibilidad y seguridad que la protección con mochila dedicada.

Desactivación

Introducción a la desactivación

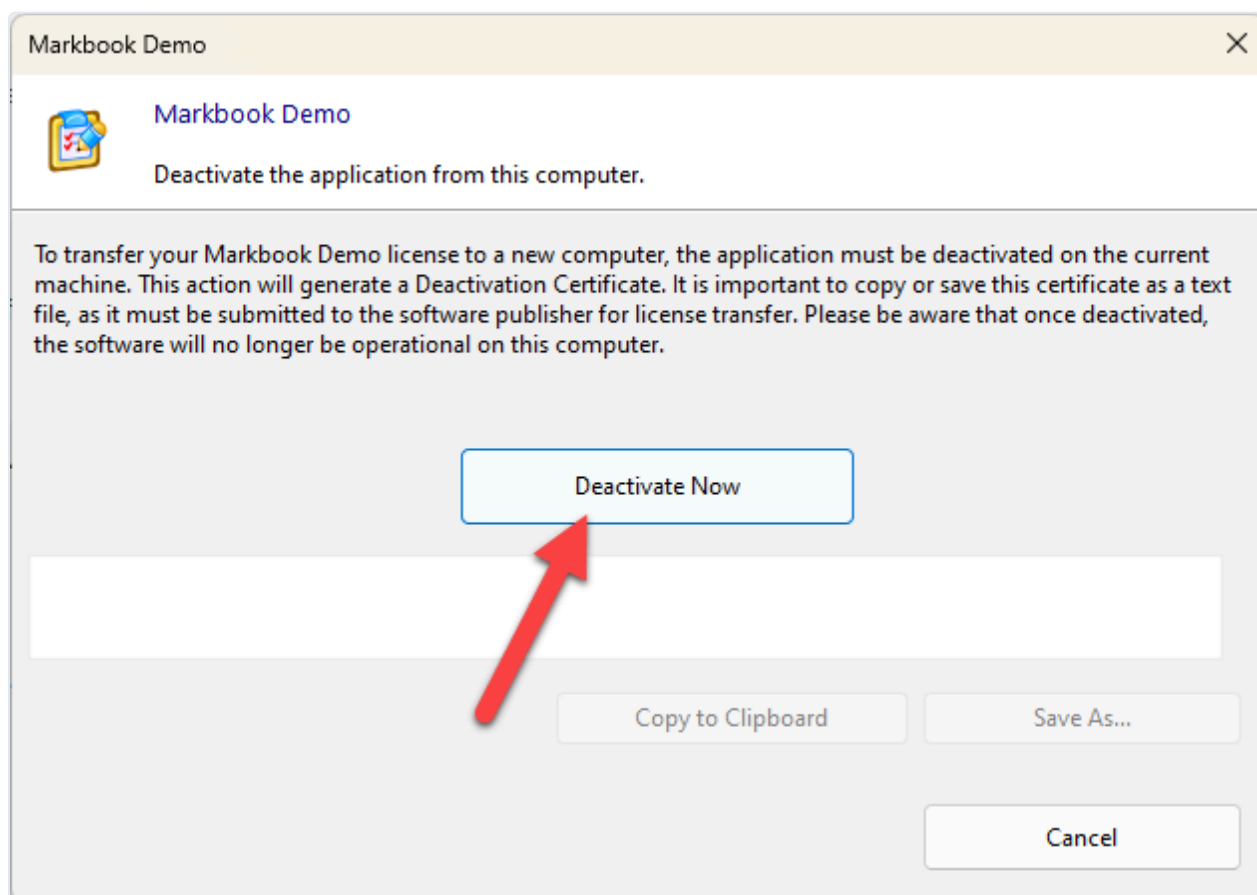
XLS Padlock ofrece un sistema de desactivación robusto que permite a los usuarios finales anular el registro de su aplicación en un equipo. Esto resulta especialmente útil cuando los clientes necesitan transferir una licencia a una máquina nueva o reactivar su software tras cambios significativos en el sistema.

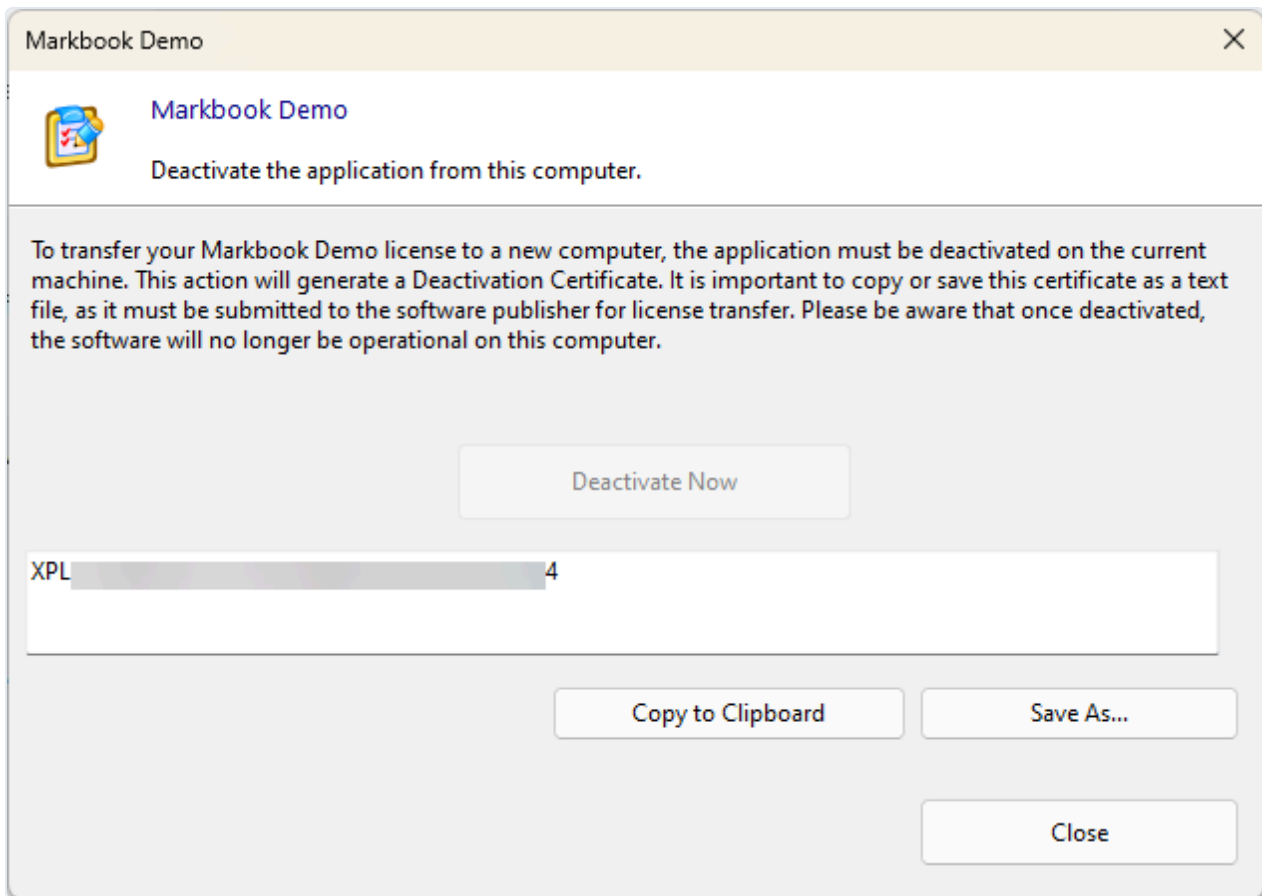
Cómo funciona la desactivación

La desactivación genera un certificado que el usuario final debe enviarle. Mediante la función **Test Deactivation Certificate** de XLS Padlock, puede verificar la autenticidad del certificado. Es fundamental tener en cuenta que, una vez que se desactiva una aplicación, su clave de activación queda invalidada. Tendrá que emitir una nueva clave al usuario para cualquier activación futura.

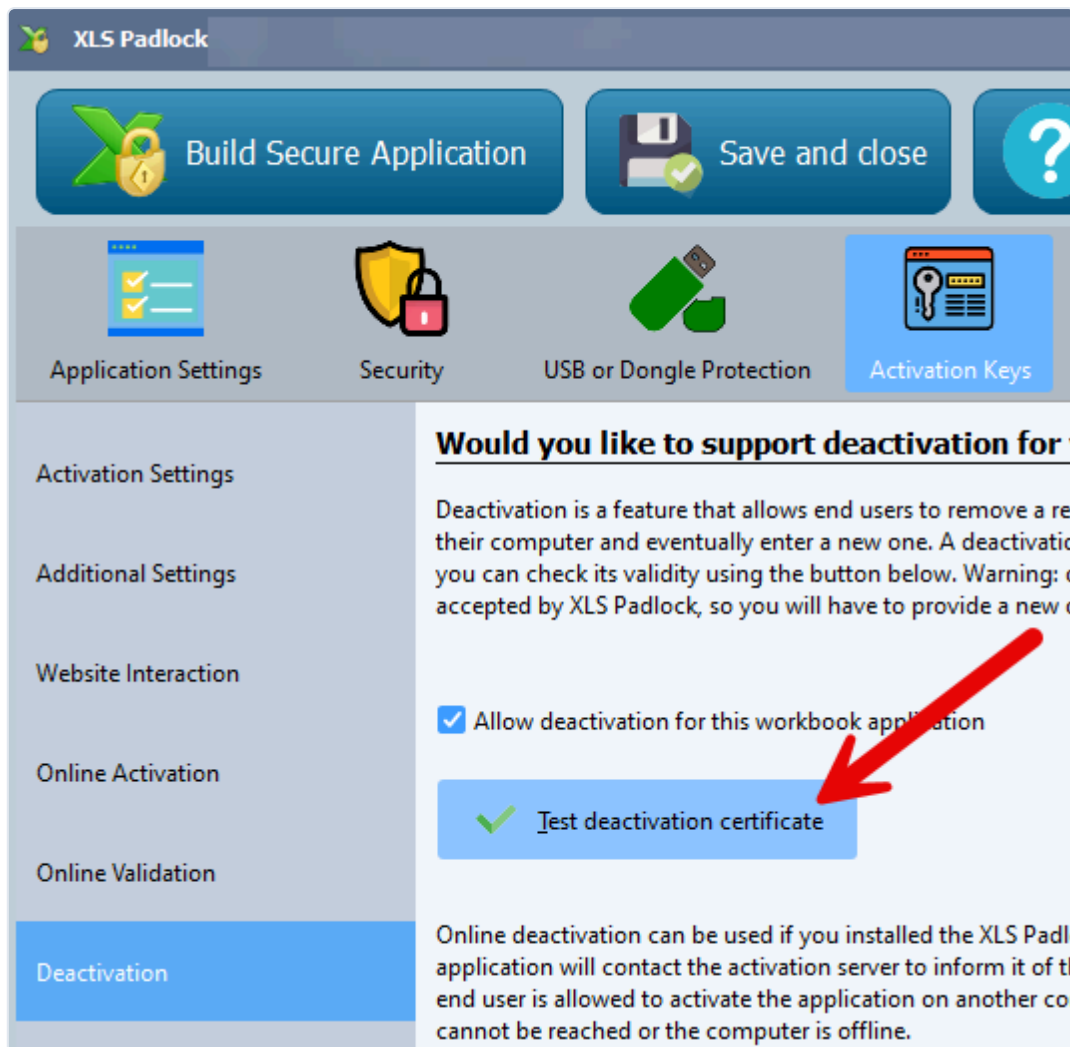
Desactivación manual

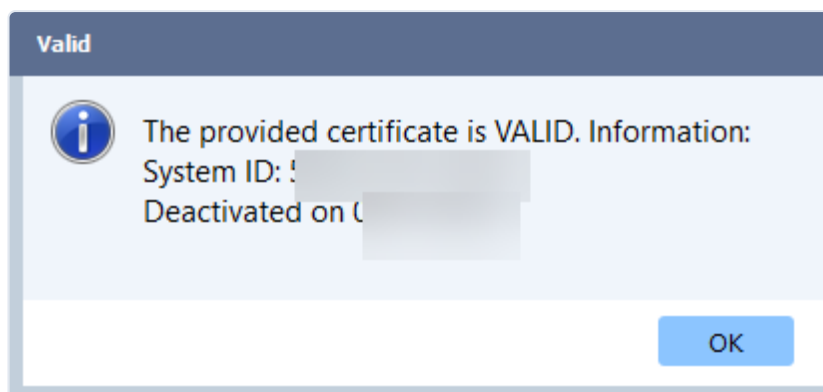
Cuando un usuario desactiva su aplicación, debe generar un certificado de desactivación y enviárselo:





Una vez recibido, puede utilizar el botón 'Test deactivation certificate' de XLS Padlock para verificar el certificado. Si la desactivación se realiza correctamente, XLS Padlock mostrará la fecha de desactivación y el identificador único del equipo del usuario:

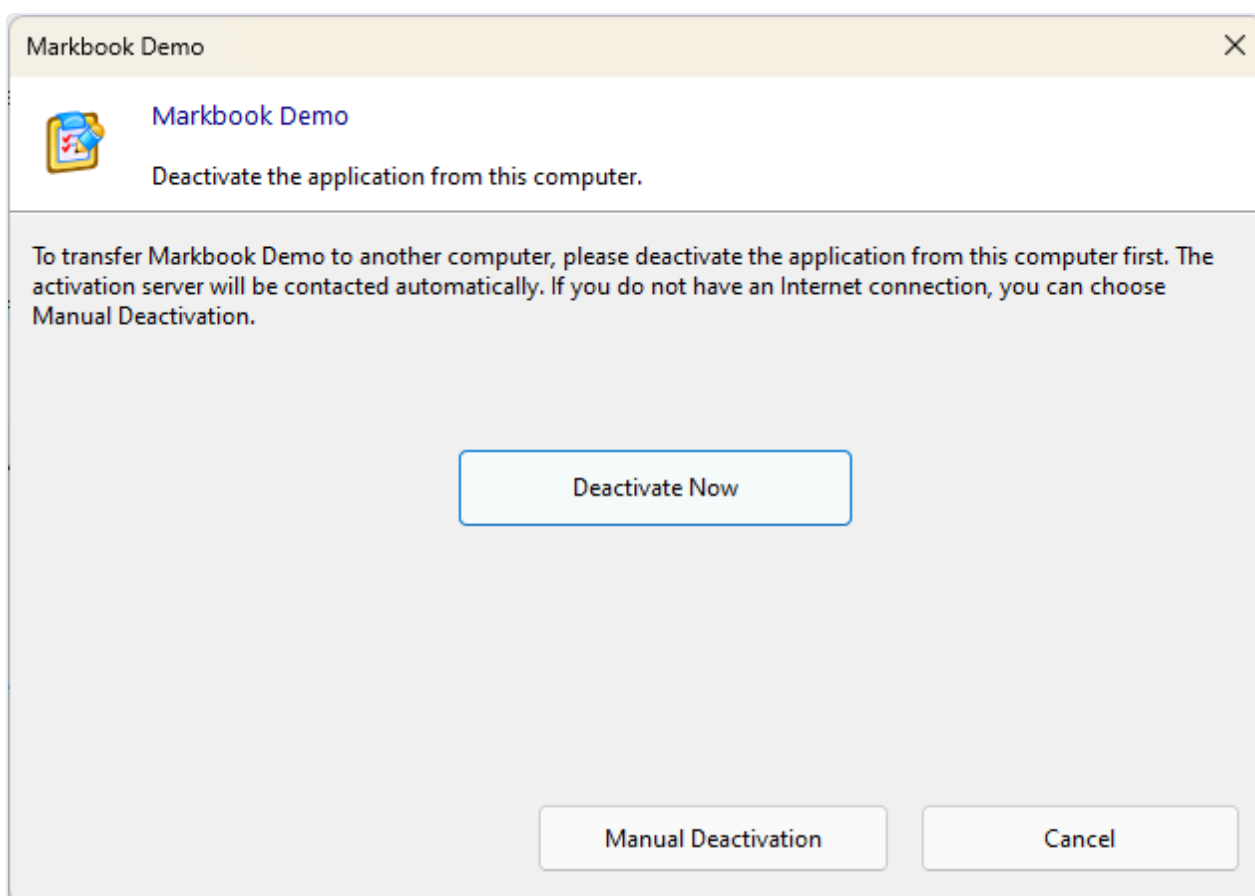




Esta información le permite gestionar nuevas activaciones para su cliente.

Desactivación en línea

Para una experiencia más fluida, XLS Padlock admite la desactivación en línea si tiene instalado el XLS Padlock Activation Kit en su servidor web. La aplicación se comunica con su servidor para anular el registro de la activación automáticamente. Si el servidor está inaccesible o el equipo está sin conexión, se crea un certificado de desactivación manual como alternativa.



La desactivación en línea automatiza todo el proceso, lo que facilita las cosas tanto para usted como para sus clientes.

Configuración de la desactivación en línea

Base Deactivation URL

Para utilizar la desactivación en línea, debe proporcionar la URL del XLS Padlock Activation Kit o del XLS Padlock WooCommerce Integration Kit en su servidor web. Por ejemplo, si instaló el kit de activación en una subcarpeta llamada "activation", la URL sería

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

Use HTTPS

Se admiten conexiones seguras mediante TLS/SSL. Debería utilizar siempre URL que comiencen por `https://` .

⚠ Si prefiere gestionar la desactivación manualmente, deje este campo en blanco.

Hide Manual Deactivation Button (in that case, it is shown if automated deactivation fails)

De forma predeterminada, la desactivación manual siempre está disponible como alternativa. Si desea animar a los usuarios a utilizar la desactivación en línea, puede optar por ocultar el botón de desactivación manual.

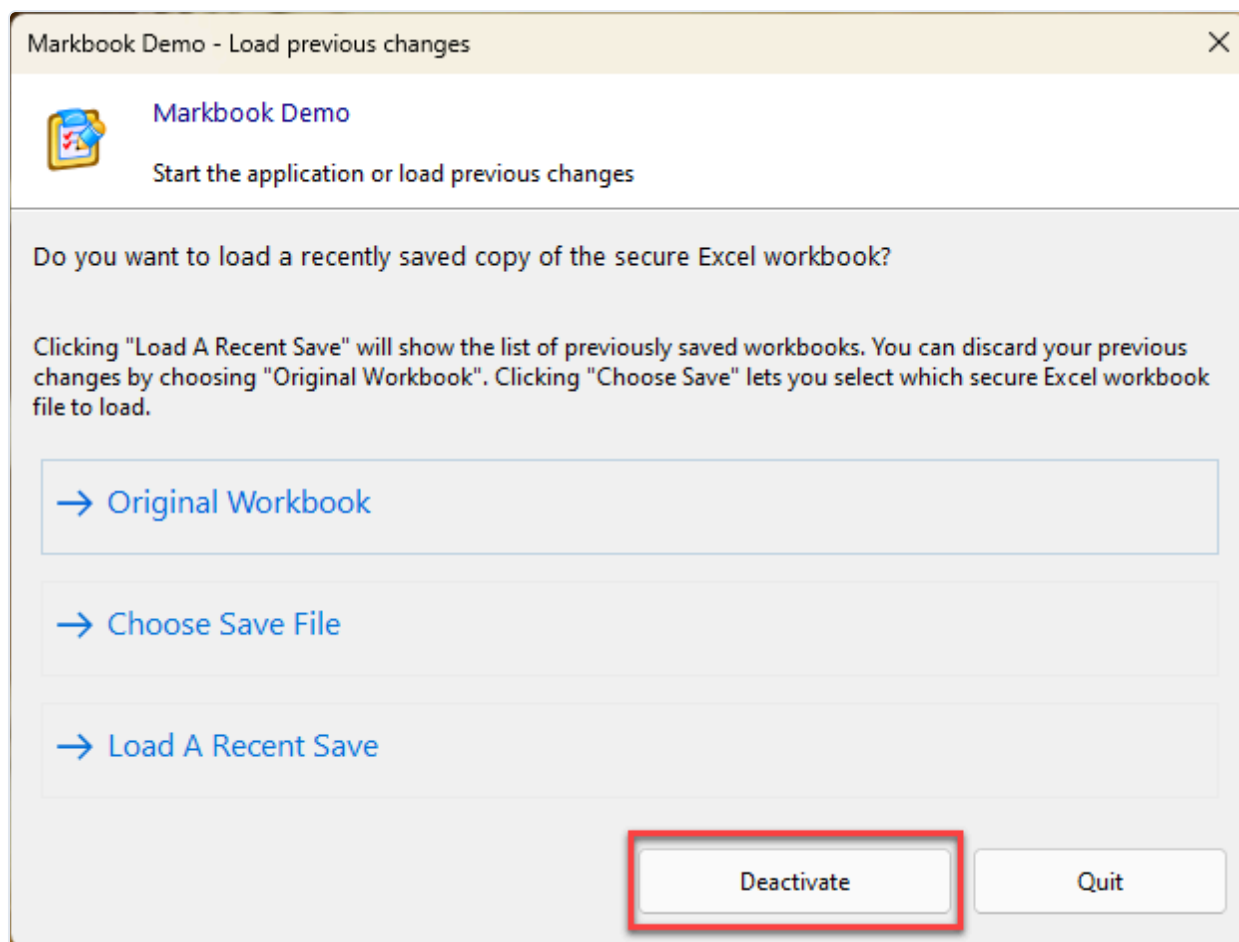
NOTA

Aunque el botón esté oculto, la aplicación ofrecerá automáticamente la desactivación manual si el proceso en línea falla (por ejemplo, si el servidor de desactivación está inaccesible o el usuario está sin conexión).

Cómo iniciar la desactivación

Un usuario puede iniciar el proceso de desactivación de tres maneras:

- La primera opción es a través del cuadro de diálogo de bienvenida integrado de la aplicación. Al iniciar la aplicación segura, aparece el cuadro de diálogo de bienvenida, que presenta diversas opciones, incluido un botón "Deactivate". Al hacer clic en este botón, los usuarios pueden comenzar el proceso de desactivación de forma sencilla y guiada:



- El segundo método implica el uso de la línea de comandos. Los usuarios avanzados pueden preferir este método para la automatización o la integración del sistema. Para ello, el usuario debe abrir el símbolo del sistema de Windows, desplazarse hasta el directorio donde se encuentra el archivo ejecutable de la aplicación segura y, a continuación, introducir el nombre del ejecutable seguido del parámetro `** -deact**`. Por ejemplo, si el ejecutable se llama `Application.exe`, el comando completo sería:

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

Al pulsar Intro, el parámetro se pasa al archivo ejecutable, lo que activa el proceso de desactivación.

- El tercer método utiliza la API de VBA proporcionada por XLS Padlock.

La desactivación es definitiva en una máquina

Tenga en cuenta que, una vez que una clave se desactiva en un equipo, queda permanentemente incluida en una lista negra en esa máquina y no se puede reutilizar. La desactivación es una acción definitiva para ese sistema concreto. Para volver a utilizar la aplicación en el mismo equipo, se requerirá una nueva clave de activación. Tenga en cuenta que esto se refiere a las claves de activación, no a los tokens de activación (tal como se definen en el WooCommerce Integration Kit).

👉 Consulte también: [Activación en línea](#chapter-online-activation), El cuadro de diálogo de bienvenida integrado

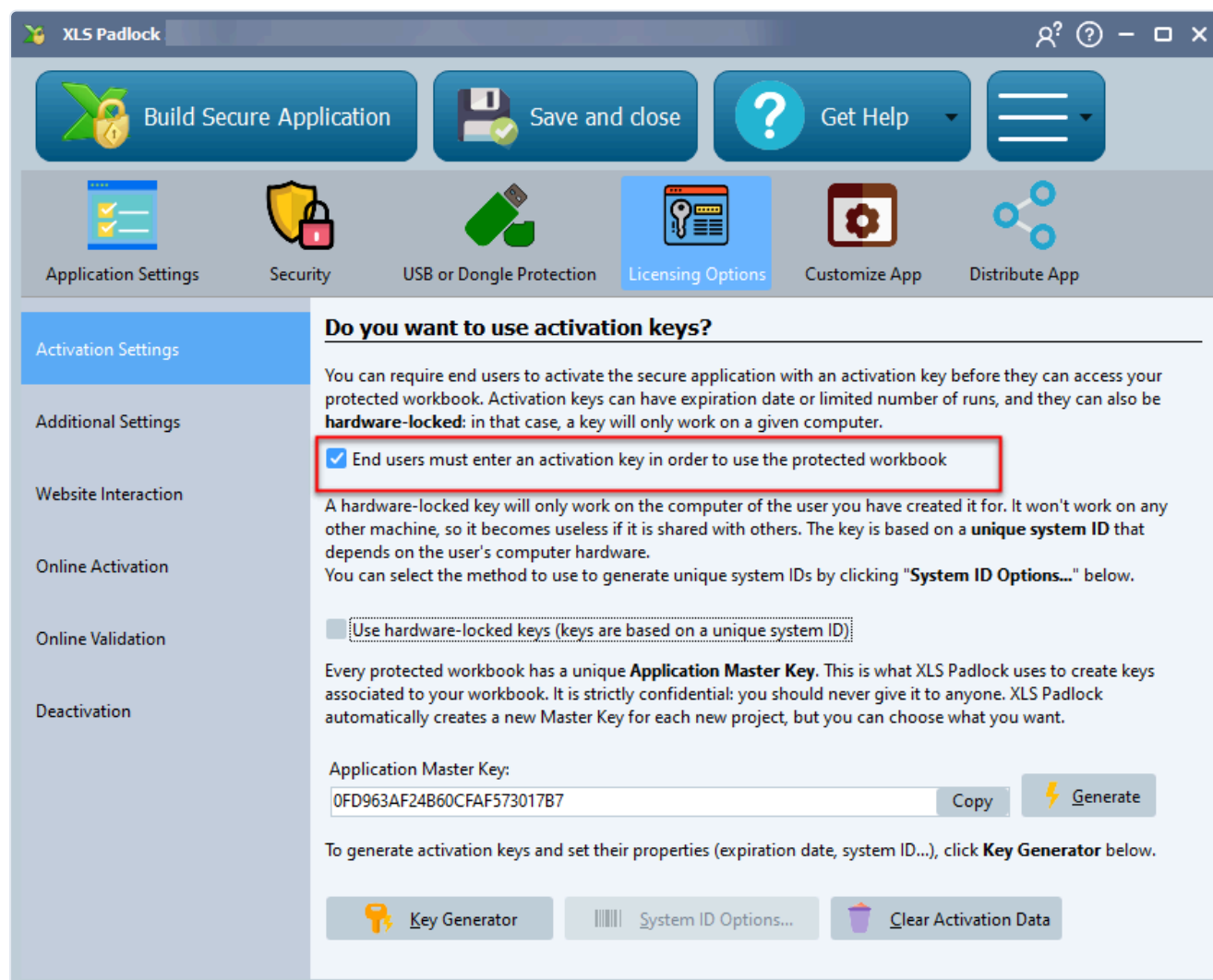
Crear libros de prueba

Puede ofrecer una vista previa de sus libros de Excel a clientes potenciales creando una versión de prueba que requiere una licencia para desbloquear toda la funcionalidad. Esto es posible con las [claves de activación de XLS Padlock](#).

Esta guía le muestra cómo configurar un periodo de prueba para su aplicación de libro.

1. Activar las claves de activación

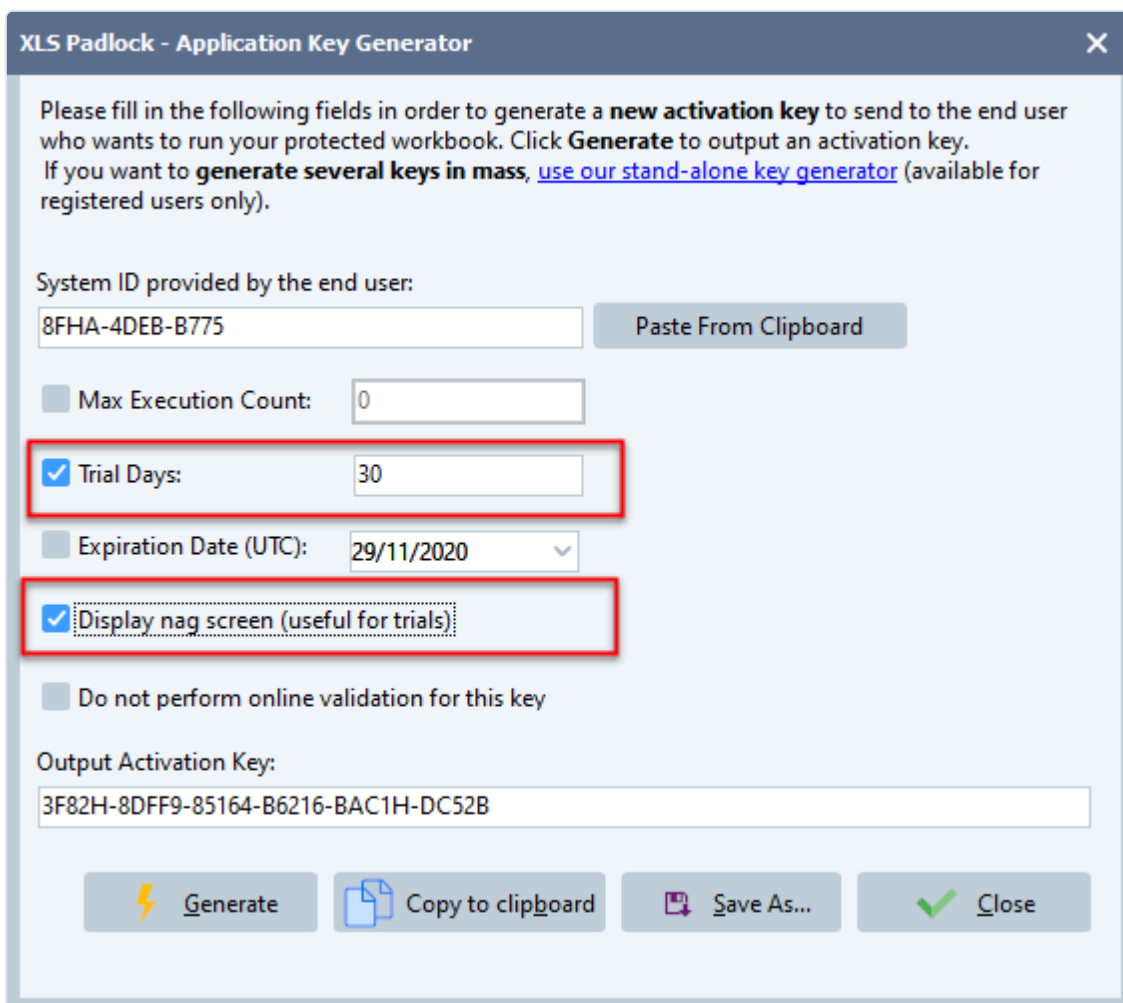
En primer lugar, active la opción **"End users must enter an activation key in order to use the protected workbook"** y recompile su aplicación.



2. Generar una clave de activación de prueba

Abra el [Key Generator](#) en XLS Padlock.

Para definir un periodo de prueba, puede establecer un límite de **Trial days** (por ejemplo, 15 días), una **Expiration Date** fija o un **Max Execution Count**.



XLS Padlock - Application Key Generator [X]

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 [v]

Display nag screen (useful for trials)

Do not perform online validation for this key

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

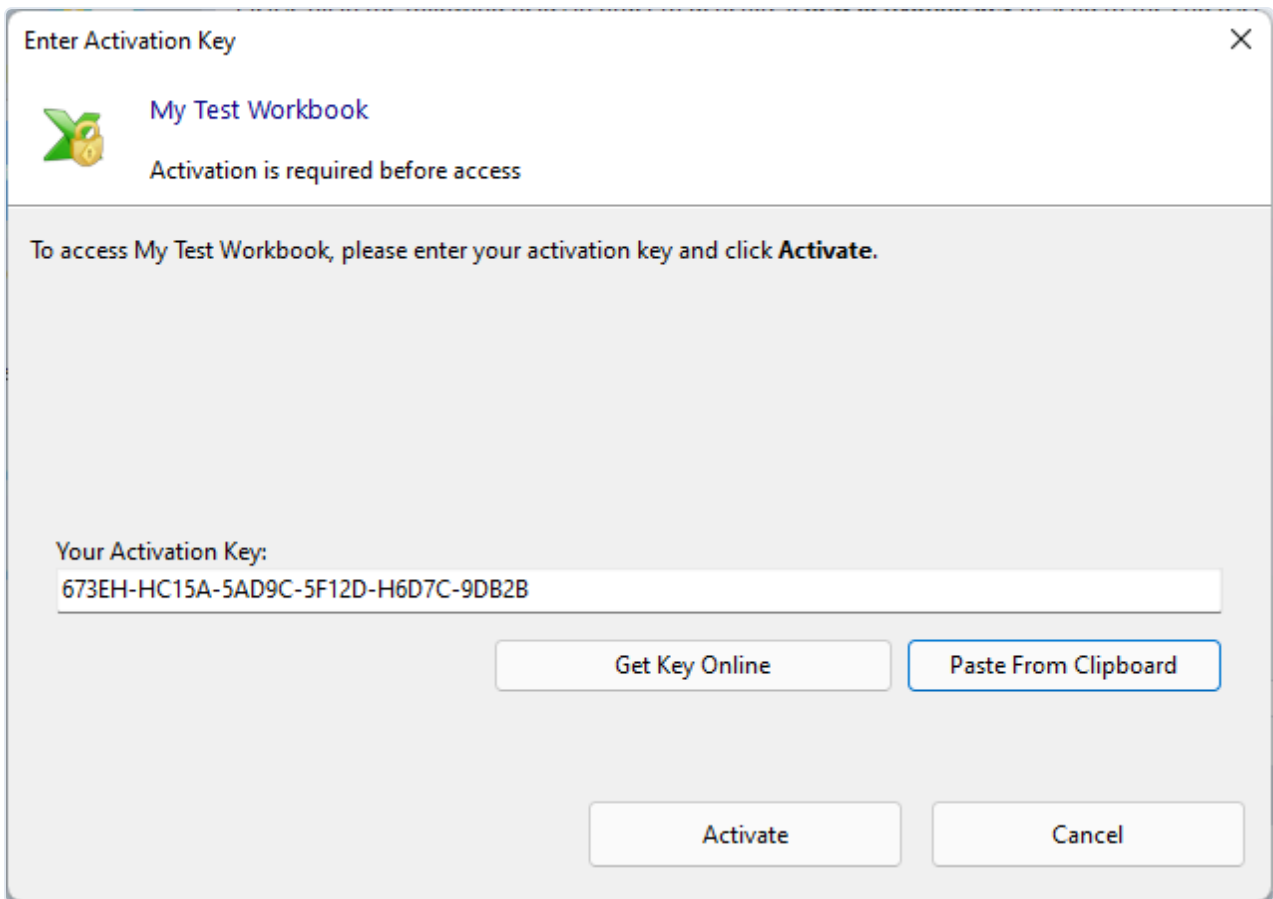
[Generate] [Copy to clipboard] [Save As...] [Close]

Para recordar a los clientes que están utilizando una versión de prueba, marque **"Display nag screen"**. Esto mostrará un cuadro de diálogo recordatorio cada vez que se inicie la aplicación.

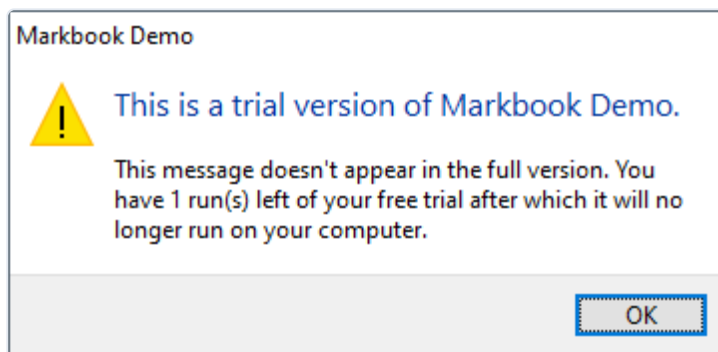
Haga clic en **Generate** para crear la clave de prueba, que luego podrá distribuir a sus usuarios.

Cómo funciona la aplicación en modo de prueba

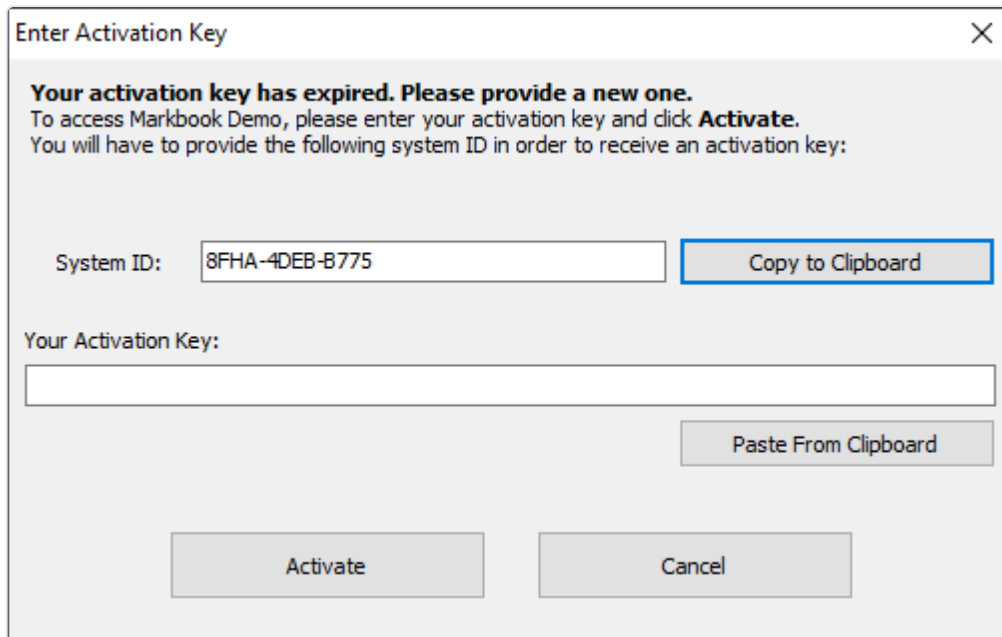
Sus clientes introducirán la clave de prueba para activar la aplicación.



Una vez activado, el libro se abrirá, pero aparecerá la pantalla recordatoria para mostrar la duración restante de la prueba.



Cuando finalice el periodo de prueba, la aplicación requerirá una nueva clave de activación, que no sea de prueba, para seguir funcionando.



Comprobar el estado de prueba con VBA

También puede [comprobar mediante programación](#) si el libro compilado está en estado de prueba utilizando la API de VBA.

Cerrar tras un tiempo determinado

¿Cómo puedo impedir que un usuario deje abierta indefinidamente una versión de prueba?

Un usuario podría obtener una clave de prueba de 1 día, abrir el archivo EXE y dejarlo abierto para siempre. La aplicación seguiría funcionando. ¿Cómo se puede evitar esto?

****Solución**:** Puede usar VBA para configurar un temporizador con `Application.OnTime` que cierre automáticamente el libro tras una duración determinada. Esta es una forma eficaz de imponer límites de prueba.

Combine esto con las opciones de seguridad de XLS Padlock para prohibir el acceso al editor de VBA y compile su código VBA para impedir que los usuarios desactiven el temporizador.

Implementación

Inserte el siguiente código en el módulo `ThisWorkbook` de su proyecto 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
```

Cómo funciona

1. **Workbook_Open** : Cuando se inicia la aplicación, este evento programa la macro `CloseAndSave` para que se ejecute al cabo de 8 horas (`TimeValue("08:00:00")`). Puede ajustar esta duración según sus necesidades (por ejemplo, `TimeValue("01:00:00")` para una hora).
2. **CloseAndSave** : Esta es la macro que realiza la acción. Guarda el libro y luego lo cierra.
3. **Workbook_BeforeClose** : Este es un paso de limpieza crucial. Si el usuario cierra el libro manualmente, este código cancela el temporizador pendiente, lo que evita que Excel intente ejecutar una macro en un libro que ya no está abierto, lo que provocaría un error.

Comprobar el estado de prueba

XLS Padlock proporciona una API de VBA que le permite determinar mediante programación si su libro compilado se está ejecutando en estado de prueba. Usando la función siguiente, puede activar o desactivar funciones específicas, mostrar mensajes de actualización o crear otras limitaciones de prueba personalizadas en función del estado de la licencia del usuario...

👉 Inserte la siguiente función en un módulo de 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
```

A continuación puede llamar a la función:

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

Advertencia

La función devolverá true solo si la clave de activación tiene activado el indicador "display nag screen" (prueba):

XLS Padlock - Application Key Generator
✕

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

System ID provided by the end user:

Paste From Clipboard

Max Execution Count:

Trial Days:

Expiration Date (UTC): ▾

Display nag screen (useful for trials)

Do not perform online validation for this key

Output Activation Key:

👉 Véase también: [Establecer restricciones en las claves de activación](#)

Días de prueba restantes

Esta función de VBA le permite **obtener mediante programación el número de días o ejecuciones restantes** de un [libro de prueba](#). También funciona con claves registradas que tienen una fecha de caducidad o un número limitado de ejecuciones.

👉 Inserte la siguiente función en un módulo de 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
```

A continuación puede llamar a la función:

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

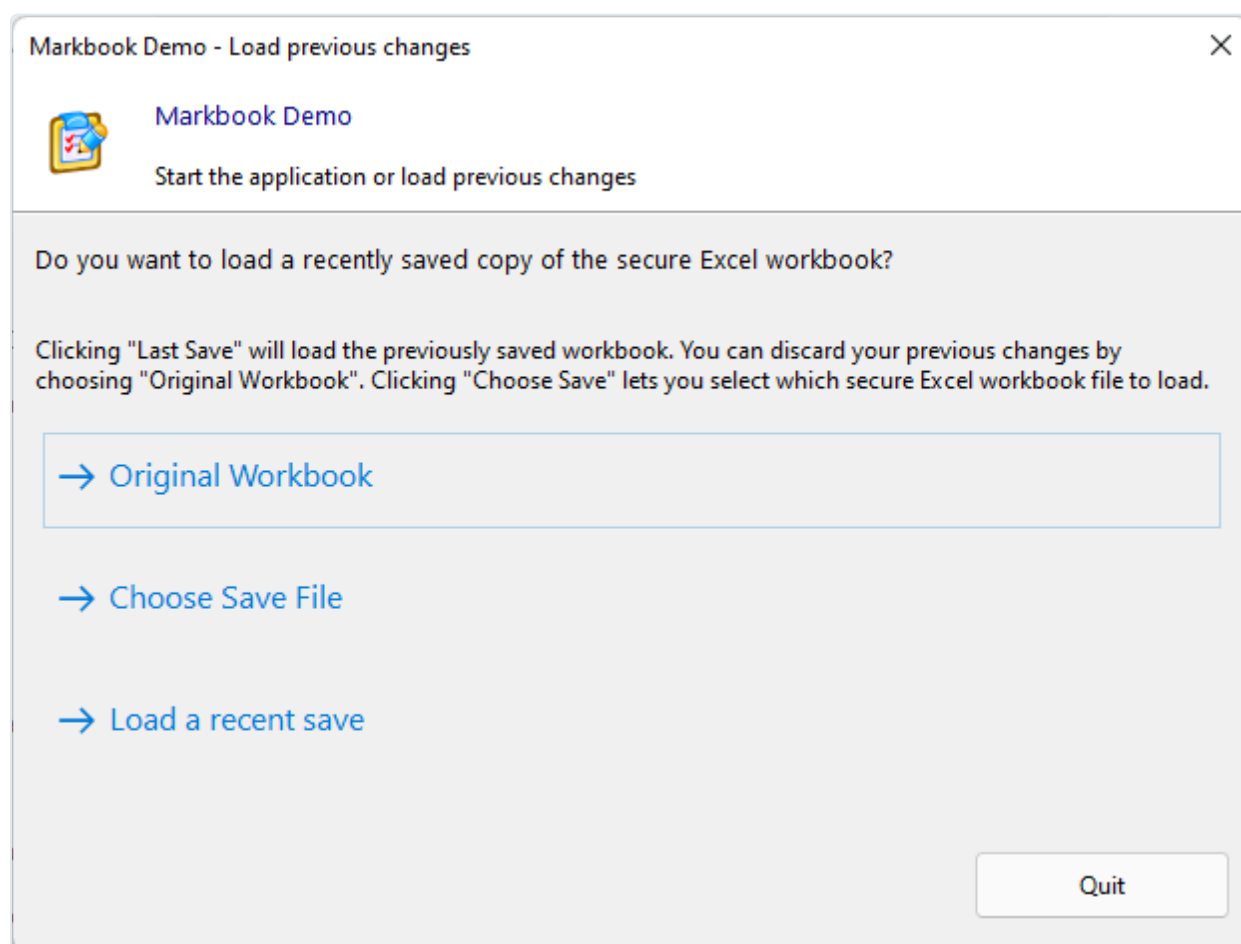
👉 Véase también: [Cómo crear libros de prueba](#)

Opciones de guardado del libro

Estas opciones le permiten ajustar con precisión cómo se guardan y cargan las modificaciones del libro cuando [habilita el guardado](#).

Always Show the "Load Previous Changes" Welcome Screen

De forma predeterminada, si no existe ningún archivo de guardado, la aplicación abre directamente el libro original. Cuando esta opción está habilitada, el cuadro de diálogo de bienvenida se muestra siempre primero, lo que permite a los usuarios empezar con el libro original o cargar un archivo de guardado existente.



Esto resulta útil si distribuye archivos de guardado predefinidos con su aplicación o si traslada el EXE a otro equipo y desea cargar un archivo de guardado existente.

Save Changes Automatically and Load Them Without Prompt

Esta opción proporciona un mecanismo de guardado automático y sin interrupciones. Los cambios del usuario se guardan en un archivo predeterminado cuando se cierra la aplicación y se vuelven a cargar automáticamente la próxima vez que se inicia. No se muestra ningún cuadro de diálogo "Save As" ni ninguna pantalla de bienvenida.

Con este modo, los usuarios no pueden crear varios archivos de guardado distintos.

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

Esta opción aumenta la seguridad al bloquear los archivos de guardado en el equipo específico donde se crearon. Un archivo de guardado creado en un PC no se puede abrir en otro, lo que impide que los usuarios compartan sus datos guardados.

➔ [Más información sobre el bloqueo por hardware de los archivos de guardado.](#)

Allow Save but Do Not Handle Loading/Saving

Habilite esta opción si desea desactivar los cuadros de diálogo de guardado integrados de XLS Padlock y gestionar usted mismo todo el proceso de guardado y carga mediante código VBA. Esto le da control total, como guardar una copia segura sin ninguna indicación al usuario.

➔ Consulte cómo [guardar una copia segura del libro sin solicitud mediante VBA.](#)

Do Not Display the "Original Workbook" Choice

Esta opción elimina la opción "Start with the original workbook" de la pantalla de bienvenida, lo que obliga a los usuarios a cargar un archivo `.XLSC` o `.XLSCE` guardado. Esto resulta útil si su aplicación siempre está pensada para usarse con archivos de datos específicos que usted proporciona.

Red de seguridad automática

Si el archivo de guardado anterior de un usuario no se puede cargar, por ejemplo, después de que usted haya cambiado la [Secret Key](#) para una nueva versión, o haya cambiado de modo de guardado entre versiones, el botón "Original Workbook" se vuelve a habilitar automáticamente en la pantalla de bienvenida para ese reintento. Esto garantiza que el usuario siempre pueda recuperar la aplicación en lugar de quedar atrapado en un bucle de mensajes de error.

👉 Véase también

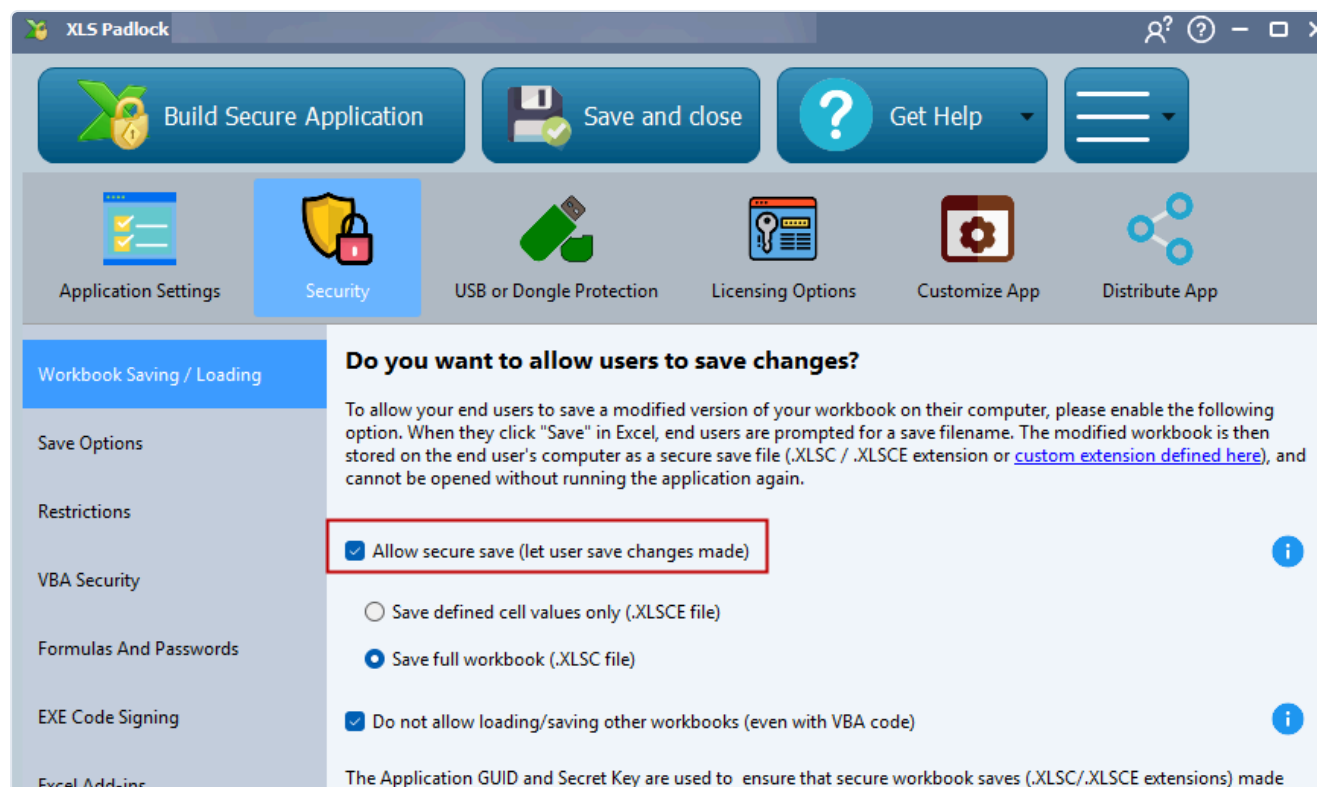
- [Descripción general del guardado y la carga de libros](#)
- [Modo de guardado: libro completo frente a valores de celda](#)
- [Recetas VBA para guardar libros](#)

Guardado y carga

Con XLS Padlock, tiene control total sobre si los usuarios pueden guardar cambios en su libro de Excel.

Cómo guarda y carga XLS Padlock los cambios del usuario

Para habilitar o deshabilitar el guardado, marque la opción "Allow secure save" en la página Security:

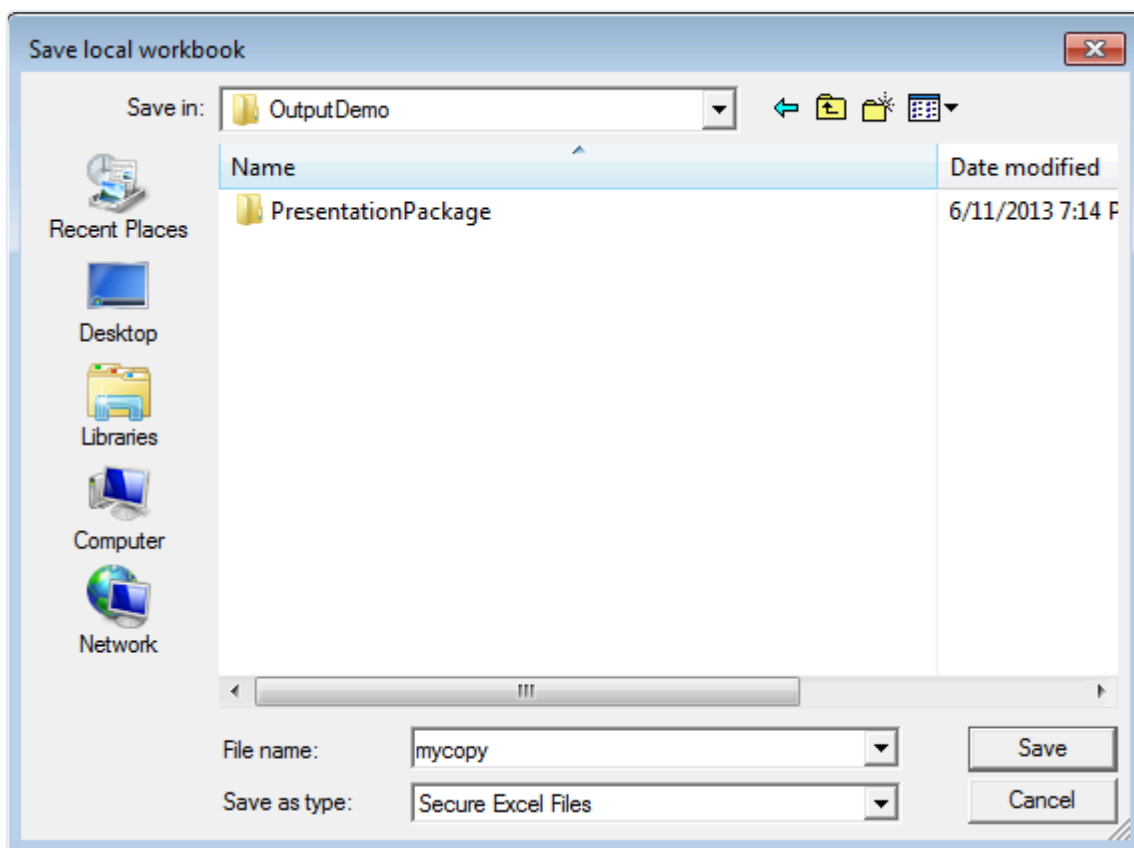


- **Si esta opción está deshabilitada**, su aplicación nunca se modificará. Cualquier cambio que realice un usuario se descarta al cerrar la aplicación. Tenga en cuenta que, en algunas versiones de Excel, el botón Guardar no se desactiva y parecerá funcionar, pero en realidad no se guarda ningún cambio.
- **Si esta opción está habilitada**, sus clientes podrán guardar sus modificaciones. A continuación, deberá [seleccionar un modo de guardado](#).

Para guardar los cambios, los usuarios finales pueden hacer clic en el botón Guardar estándar



de Excel o usar el menú "File => Save". Aparecerá entonces el cuadro de diálogo "Save As", que les preguntará dónde guardar su copia segura del libro:



A este archivo de guardado se le asigna la extensión `.XLSC` (o `.XLSCE`) y **no se puede abrir sin volver a ejecutar la aplicación segura.**

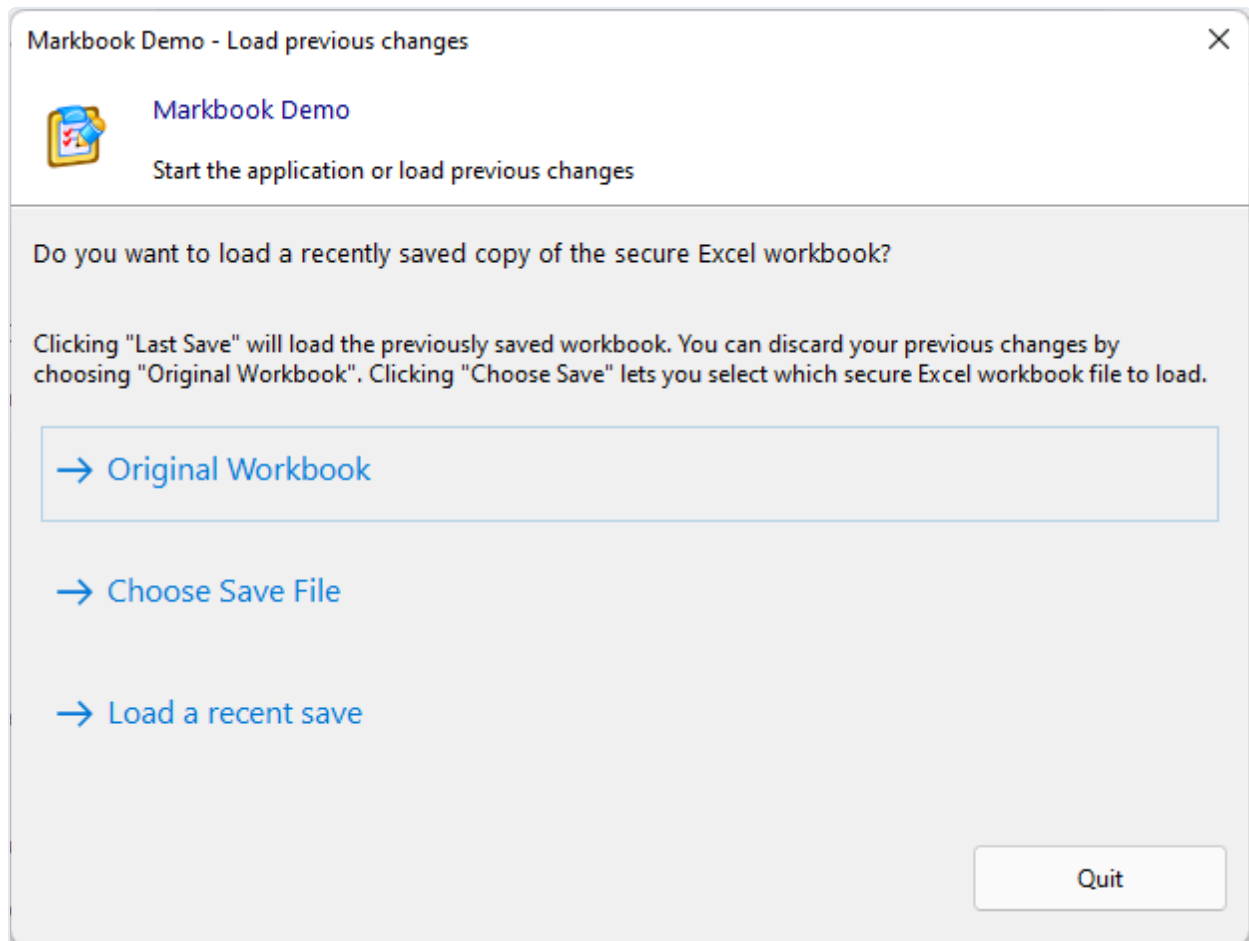
De forma predeterminada, el cuadro de diálogo preguntará a los usuarios finales si desean sobrescribir sus cambios. Puede deshabilitar este comportamiento en las [Opciones avanzadas](#).

Extensión personalizada del archivo de guardado

Puede definir su propia extensión personalizada para los archivos de guardado en las [Opciones avanzadas](#).

Cómo cargan sus cambios los usuarios finales

La próxima vez que ejecuten su aplicación, los clientes podrán elegir entre abrir su libro original o una versión que hayan guardado anteriormente. La pantalla de bienvenida les permite "Choose Save File" o cargar un archivo desde el menú "Recent Saves".



De forma predeterminada, el cuadro de diálogo "Save As" se muestra cada vez que un usuario hace clic en el botón Guardar. El último archivo guardado se recuerda automáticamente. Tenga en cuenta que solo funciona el comando `Save` ; `Save As` está siempre deshabilitado.

Véase también

- [Cómo migrar los datos del usuario desde una versión anterior con VBA](#)
- [Elegir cómo se guardan los datos en el libro](#)
- [Guardar y restaurar valores personalizados mediante programación con código VBA](#)

Modo de guardado: completo o valores de celdas

Si permite que los usuarios guarden su trabajo, debe definir cómo guardará y cargará XLS Padlock esos cambios. XLS Padlock ofrece dos modos de guardado, que se pueden seleccionar en la página [Guardado y carga del libro](#).



Save Full Workbook (archivo .XLSC)

En este modo, se almacena una copia completa y cifrada del libro, incluidos todos los cambios del usuario, en un archivo seguro `.XLSC`. Este archivo solo lo puede abrir su aplicación protegida, lo que garantiza que el libro permanezca seguro.

El **Full Save mode** cifra y guarda todo el libro exactamente como está en el momento del guardado. Esto significa que, si más adelante distribuye un EXE actualizado, los usuarios que abran un archivo de guardado antiguo verán su trabajo guardado previamente, no sus nuevas actualizaciones. Su archivo de guardado es una instantánea completa del libro en ese momento.

Los archivos de guardado completos se pueden compartir entre usuarios, a menos que los [bloquee a un equipo concreto](#). También puede [descifrar usted mismo estos archivos de guardado](#) para recuperar los datos del usuario.

Cuándo usar el modo Full Save

El modo Full Save es la opción predeterminada para la mayoría de los usuarios y se recomienda especialmente para libros complejos en los que los usuarios realizan cambios extensos en varias hojas.

El principal inconveniente es que los archivos de guardado del usuario no son automáticamente compatibles con las versiones actualizadas de su aplicación. Si publica con frecuencia actualizaciones de la lógica o el diseño de su libro, considere usar el modo Cell Values en su lugar.

👉 Véase también: [Cómo migrar los datos del usuario desde una versión anterior con VBA](#)

Save Defined Cell Values Only (archivo .XLSCE)

En este modo, solo se guardan y restauran los valores de celdas específicas y predefinidas. Esta es la opción ideal si actualiza con frecuencia su libro de origen y distribuye nuevas versiones.

Cuando un usuario carga su archivo de guardado `.XLSCE` con su nuevo EXE, sus datos guardados se cargan en la estructura de libro actualizada. Esto les permite beneficiarse de sus actualizaciones sin perder su trabajo.

La principal desventaja es que debe [definir manualmente qué celdas guardar](#) antes de compilar su aplicación.

Al igual que los archivos `.XLSC`, los archivos `.XLSCE` están cifrados de forma segura y solo los puede abrir su aplicación. Esto se aplica mediante una [clave secreta única](#) específica de su proyecto.

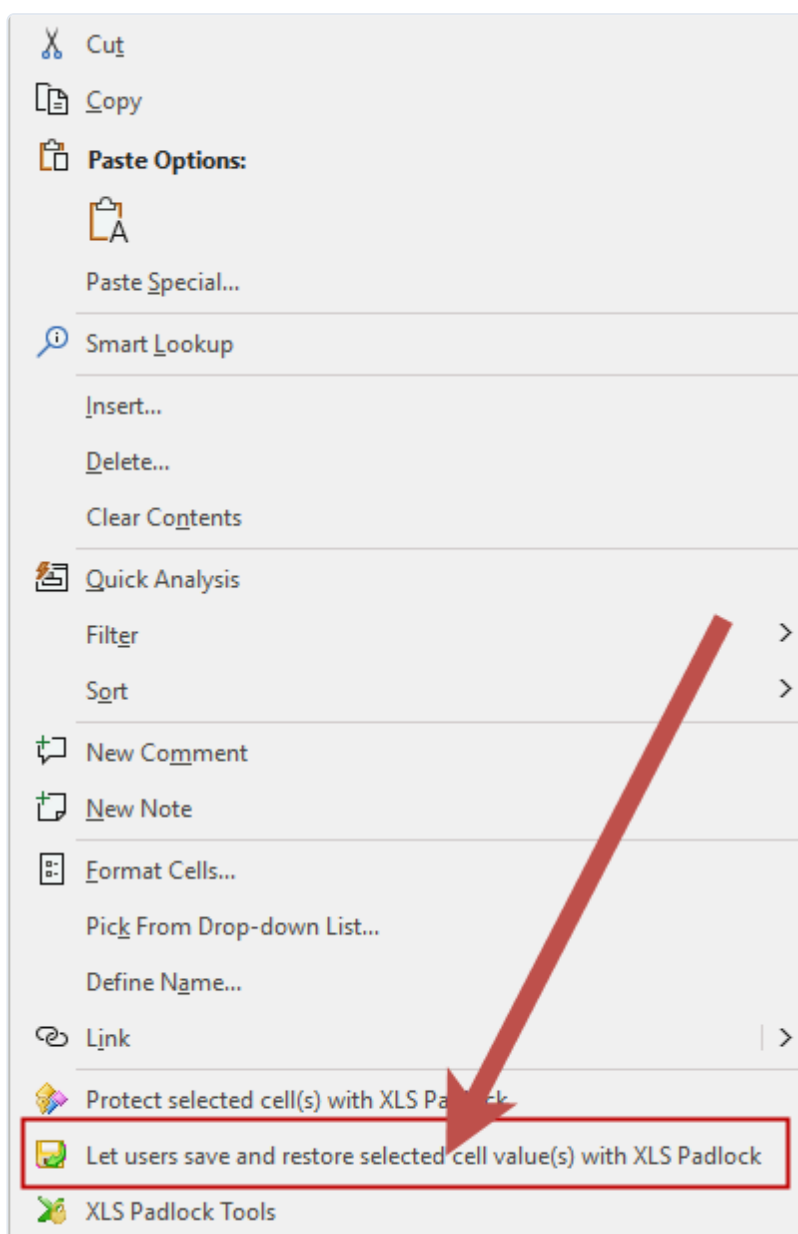
Elija su modo de guardado de antemano

Los formatos Full Save (`.XLSC`) y Cell Values (`.XLSCE`) no son intercambiables. Una vez que haya distribuido una aplicación usando un modo, cambiar al otro en una versión posterior significa que los archivos de guardado de los usuarios existentes no se cargarán, XLS Padlock recurrirá de forma silenciosa al libro original incrustado para esos usuarios en el primer inicio de la nueva versión, y se perderán todos los datos que hubieran guardado en el modo anterior. Elija al principio del proyecto el modo que se ajuste a su estrategia de actualización y manténgalo.

Definir celdas para guardar y restaurar

Al usar el modo **Save defined cell values only**, debe especificar qué celdas guardará y restaurará su aplicación.

👉 Para ello, simplemente **haga clic con el botón derecho en una o varias celdas** y elija **"Let users save and restore selected cell value(s) with XLS Padlock"** en el menú contextual.

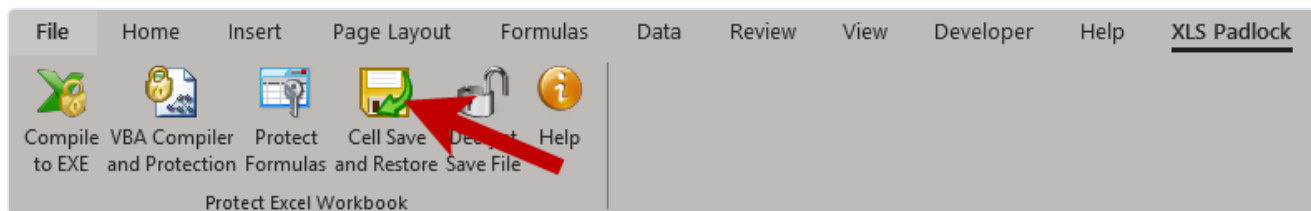


XLS Padlock confirmará que las celdas seleccionadas están marcadas para guardarse y restaurarse.

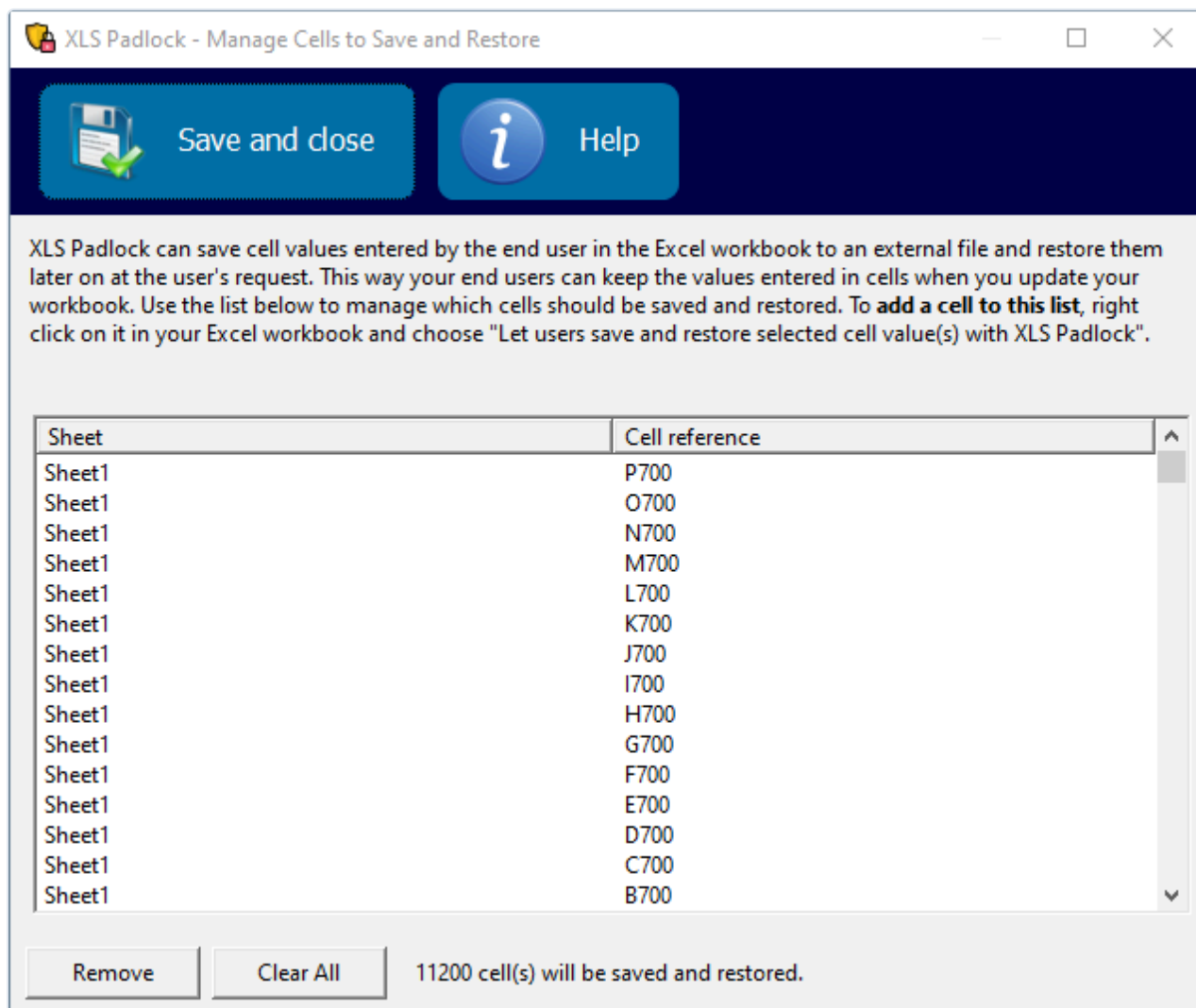
También puede [almacenar y restaurar valores personalizados con código VBA](#).

Resumen de las celdas que se guardarán

Para ver todas las celdas configuradas para guardarse, haga clic en **"Cell Save And Restore"** en la pestaña de la cinta de XLS Padlock:



Esto abre una ventana que muestra todas las celdas configuradas. Desde aquí, puede eliminar celdas individuales o borrar toda la lista.



Restaurar y guardar con VBA

Al usar el modo **Save defined cell values only**, XLS Padlock le permite guardar y restaurar valores personalizados además de los valores de celda predefinidos. Esto resulta útil para guardar variables, ajustes o datos que no se almacenan directamente en una celda.

Resumen

Esta funcionalidad se basa en dos eventos de VBA y dos funciones de la API de VBA. Coloque el código en las subrutinas de evento para leer o escribir sus datos personalizados cuando el usuario cargue o guarde su trabajo.

Eventos y funciones principales

Eventos de VBA

Las dos subrutinas de evento siguientes deben colocarse en un módulo de su libro de Excel. XLS Padlock las llamará automáticamente durante el proceso de guardado/carga.

```
' 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
```

Funciones de la API de VBA

- **WriteCustomCellValue(UniqueID, Value)**: escribe un único valor de cadena asociado a un ID único.
- **ReadCustomCellValue(UniqueID, DefaultValue)**: lee un único valor de cadena para un ID determinado. Si no se encuentra el ID, devuelve el `DefaultValue`.

Importante

`WriteCustomCellValue` y `ReadCustomCellValue` solo funcionarán cuando se llamen desde dentro de los eventos `XLSPadlock_SaveCustomValues` y `XLSPadlock_RestoreCustomValues`, respectivamente.

Lectura y escritura de valores individuales

Así es como escribiría un valor individual:

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

Y así es como lo leería de nuevo:

```
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
```

Lectura de todos los valores como un diccionario

También puede leer todos los valores personalizados guardados a la vez pasando una cadena vacía como ID a `ReadCustomCellValue`. Esto devuelve un objeto `Scripting.Dictionary`.

Referencia necesaria

Para usar esta función, debe añadir una referencia a **“Microsoft Scripting Runtime”** en su proyecto de VBA (desde el menú del VBE, vaya a 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

```
## Ejemplo completo: guardar/restaurar una columna
```

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

Todo el código siguiente debe colocarse en un único módulo.

```
</div>
```

Esta función auxiliar genera una cadena separada por comas a partir de los valores de un rango

```
`` `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
```

Este evento se llama cuando el usuario guarda su trabajo. Utiliza la función auxiliar para convertir todo el rango usado de la columna A en una única cadena y la guarda.

```
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
```

Este evento se llama cuando el usuario carga un archivo guardado. Lee la cadena y restaura los valores de nuevo en la columna 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
```

Consulte también

- [Cómo migrar los datos de usuario de una versión anterior](#chapter-excel-vba-migrate-user-da

Acceder al libro seguro y a los archivos complementarios

NOTA

Esta guía explica cómo iniciar una nueva instancia de Excel desde su libro protegido y concederle acceso al libro seguro y a sus archivos complementarios.

De forma predeterminada, solo la instancia de Excel iniciada por su EXE compilado puede acceder al libro seguro y a sus archivos asociados. Si crea una nueva instancia de Excel mediante VBA, esta no tendrá permiso para abrir estos archivos protegidos.

Para conceder el acceso, debe usar la API de XLS Padlock para pasar el identificador de ventana (HWND) de la nueva instancia de Excel a la aplicación principal. Esto autoriza a la nueva instancia a trabajar con el sistema de archivos virtual.

```
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

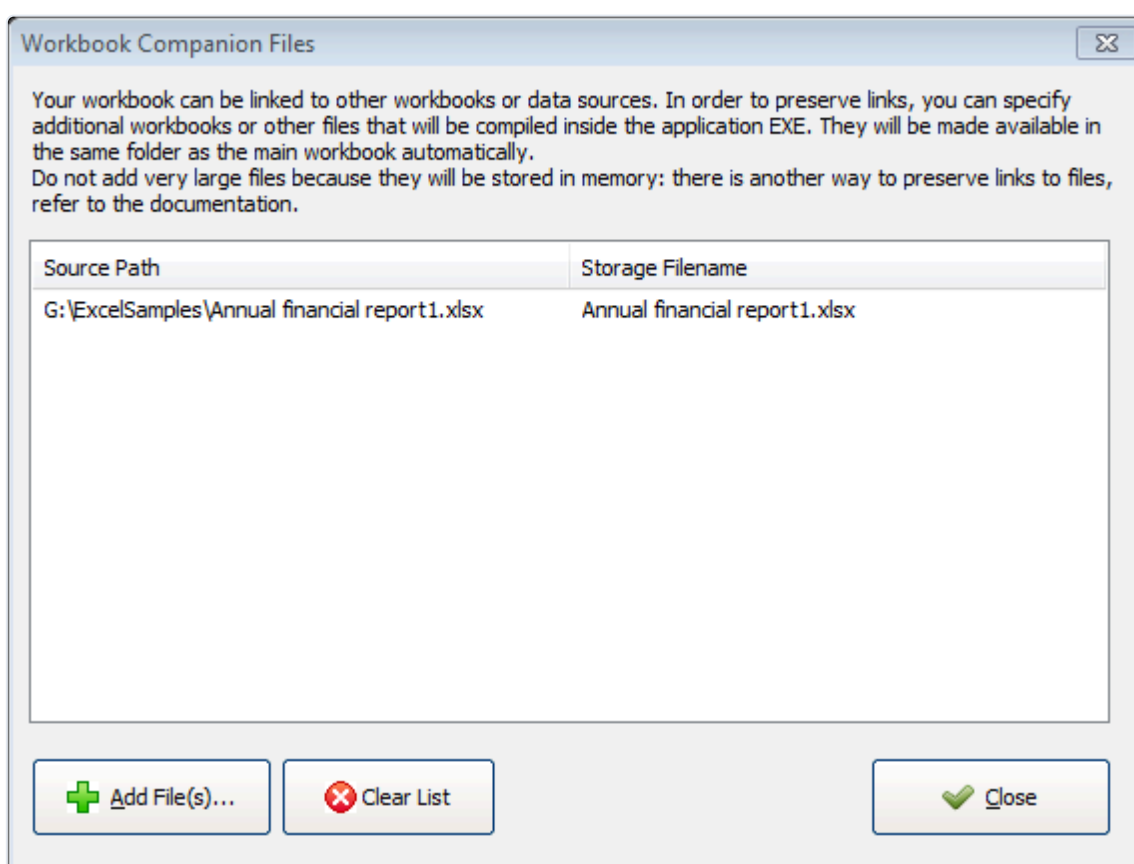
Public Function GetPathToFileInEXEFolder(ByVal Filename As String) As String
' Helper function to get the full path to a file in the EXE's directory.
' See: https://www.xlspadlock.com/es/doc/get-the-path-to-a-file-in-the-same-folder-as-the-c
Dim XLSPadlock As Object
On Error Resume Next
Set XLSPadlock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

Añadir archivos complementarios

Si su libro está vinculado a otros libros o fuentes de datos, puede añadirlos como "archivos complementarios". Esto los integra directamente en el EXE de la aplicación principal, lo que facilita la distribución. Cuando la aplicación se ejecuta, estos archivos quedan disponibles en la misma carpeta virtual que el libro principal.

Incluso puede incluir complementos XLL.

Por ejemplo, si su libro utiliza un archivo de texto externo como fuente de datos, añadirlo como archivo complementario garantiza que siempre se encontrará cuando el libro compilado se abra en cualquier equipo.



Para añadir archivos, haga clic en **Add Files** y selecciónelos. Aparecerán en la lista. La columna "Storage Filename" muestra el nombre de archivo que se utilizará cuando el archivo se compile en el EXE y se restaure en tiempo de ejecución.

Evite los archivos grandes

No añada archivos muy grandes, ya que se almacenarán por completo en memoria. Para vincular otros libros o archivos de datos de gran tamaño, es preferible colocarlos en la misma carpeta que el EXE de su aplicación y referenciarlos mediante la función `PLEvalVar("XLSPATH")`. Consulte [Usar referencias externas e hipervínculos](#) para obtener más detalles.

Acceder a archivos complementarios con VBA

Para acceder a los archivos complementarios con VBA, puede utilizar la siguiente función:

```
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
```

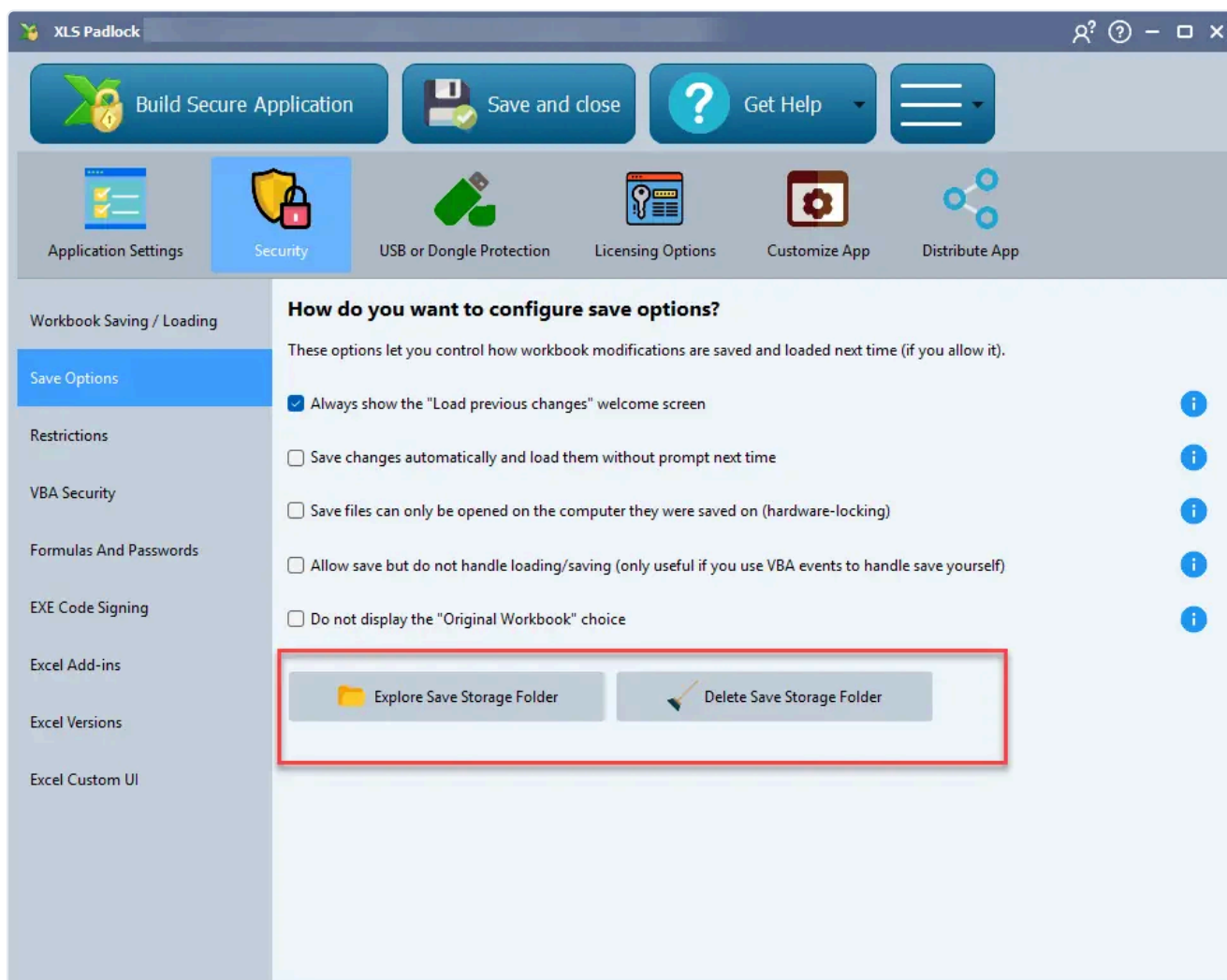
Después puede utilizar esta función para abrir un archivo complementario:

```
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
```

Carpeta de almacenamiento de guardados

Los botones **Explore Save Storage Folder** y **Delete Save Storage Folder** proporcionan acceso rápido a la carpeta local de su PC donde su aplicación protegida almacena sus datos.



Esta carpeta contiene:, Archivos guardados automáticamente (si esta función está habilitada)., El archivo de historial de guardado., Datos de licencia y activación.

Si el modo portátil está habilitado (en [Opciones avanzadas](#)), estos botones abrirán la misma carpeta que contiene el archivo EXE de la aplicación. Por seguridad, el botón "Delete" está deshabilitado en este modo.

Almacenar los cambios dentro del EXE

No, los usuarios no pueden guardar los cambios directamente en el archivo `.exe` compilado. Modificar un archivo ejecutable después de la compilación puede corromperlo y, a menudo, provoca falsos positivos en el software antivirus.

Por este motivo, XLS Padlock está diseñado para guardar los datos del usuario en archivos de guardado cifrados independientes (con extensiones `.xlsc` o `.xlsce`), no de nuevo en el EXE de la aplicación principal.

Considere su EXE compilado como la propia aplicación (como `Excel.exe`), y los archivos de guardado como los documentos con los que trabaja.

Restringir la carga y el guardado

Una posible vulnerabilidad es que los libros de Excel se pueden guardar en disco mediante comandos VBA u OLE. Para impedir este método de extracción de datos, XLS Padlock incluye una potente opción de seguridad denominada **“Do not allow loading/saving other workbooks”** (No permitir cargar/guardar otros libros).

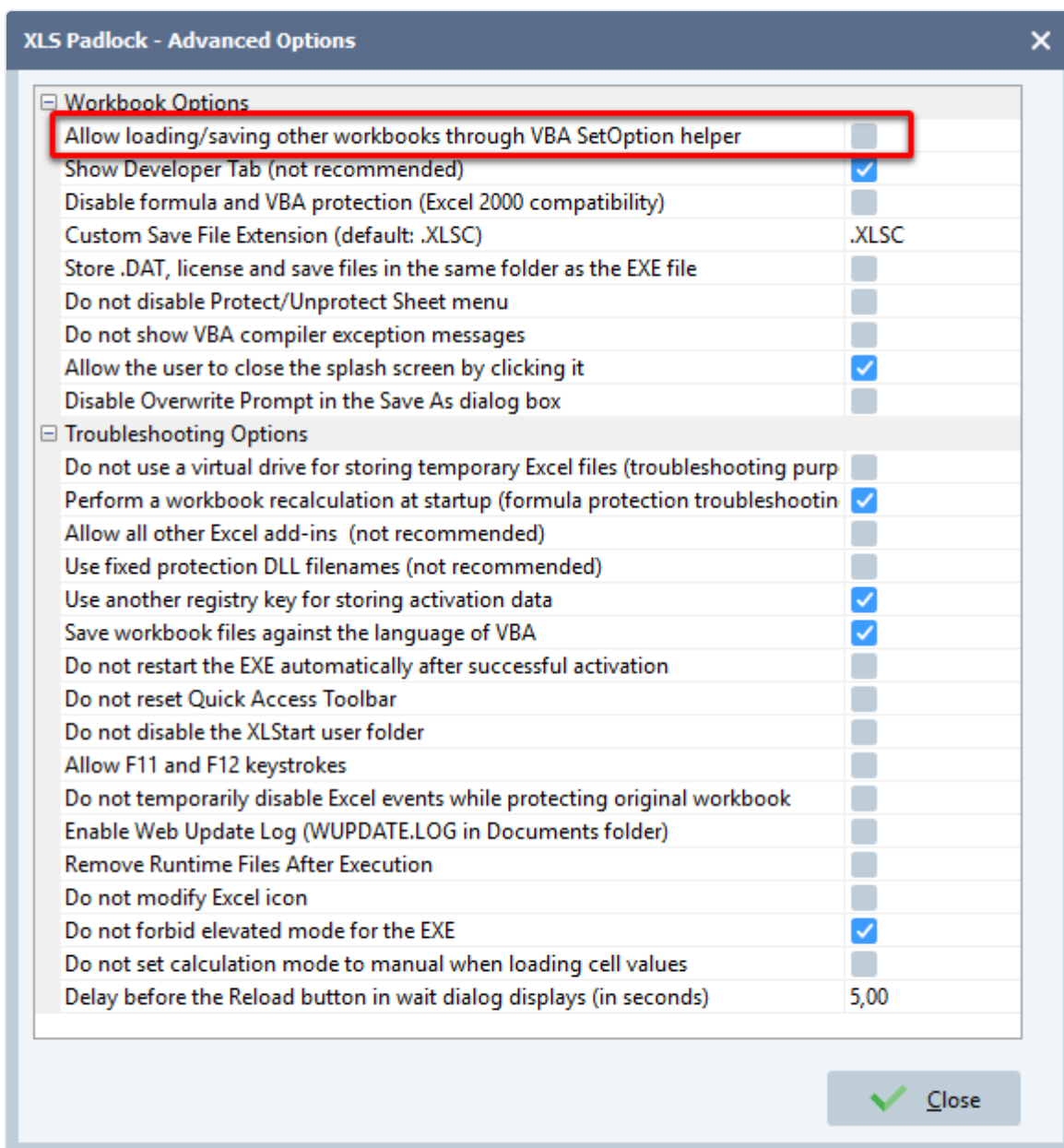
Cuando está habilitada, esta función hace que la instancia de Excel que ejecuta su aplicación sea **incapaz de cargar o guardar cualquier otro archivo de libro**. Esto bloquea de forma efectiva que los usuarios abran otros libros e impide los hacks basados en VBA que intentan copiar datos a un nuevo archivo sin protección.



¿Qué sucede si mi aplicación necesita guardar otros libros? La solución VBA SetHelper

Esta función de seguridad también bloquea el código VBA estándar para guardar o cargar libros. Si su aplicación necesita realizar legítimamente estas acciones, existe una solución alternativa.

Puede deshabilitar temporalmente la restricción dentro de su código VBA. Para hacerlo, primero debe habilitar la opción avanzada **"Allow loading/saving other workbooks through VBA SetOption helper"** (Permitir cargar/guardar otros libros mediante el helper VBA SetOption) aquí:



A continuación, puede usar el siguiente código VBA:

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

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

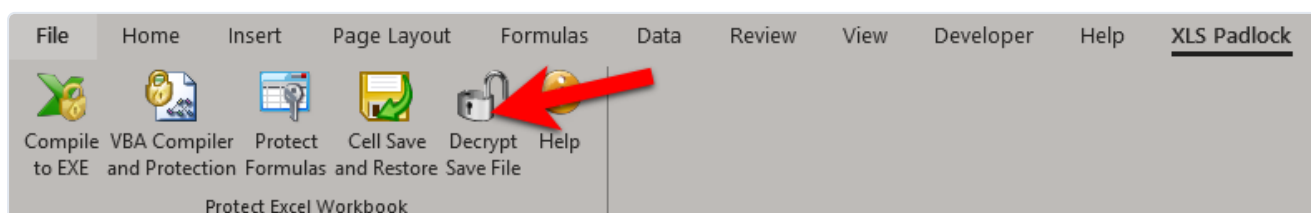
' Your code to save a normal workbook
ActiveWorkbook.SaveAs "D:\My Documents\NormalWorkbook.xlsx"

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

Abrir y descifrar los guardados

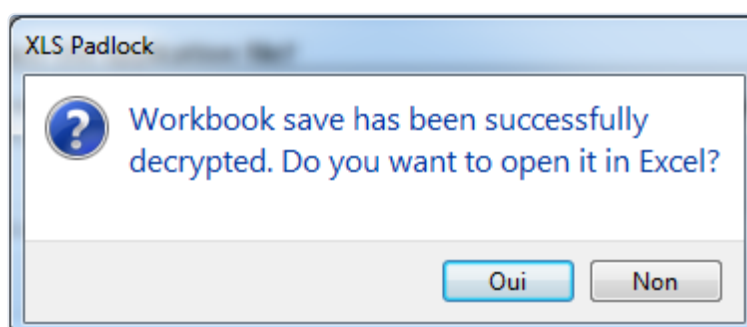
Los archivos de guardado creados por su aplicación están cifrados y no se pueden abrir directamente en Excel. Sin embargo, como autor original del libro, **puede descifrar cualquier archivo de guardado generado por su aplicación**. Esto resulta útil para recuperar datos si un cliente le envía su archivo de guardado.

Para abrir un archivo de guardado (`.xlsc` o `.xlsce`), elija **“Decrypt Save File”** (Descifrar archivo de guardado) en la cinta de XLS Padlock en Excel:



Descifrar archivos `.XLSC` (modo de guardado completo)

Si selecciona un archivo `.xlsc` creado con el [modo de guardado completo](#), XLS Padlock descifrará el libro completo y lo abrirá.



Limitaciones del descifrado

- Las fórmulas protegidas con XLS Padlock no se pueden recuperar al descifrar un archivo de guardado. Es posible que el libro descifrado no sea totalmente funcional. Esta función solo debe usarse para recuperar los datos introducidos por el usuario.
- La misma limitación se aplica al proyecto VBA si lo ha bloqueado o compilado.
- Para usar el libro con todas las funciones intactas, debe ejecutar su aplicación segura y cargar el archivo de guardado con normalidad.

Descifrar archivos `.XLSCE` (modo de valores de celda)

Si selecciona un archivo `.xlsce` creado con el modo de guardado de valores de celda, XLS Padlock le preguntará si desea sobrescribir las celdas de su libro actualmente abierto con los valores del archivo de guardado.

Esto sobrescribirá sus datos

Confirmar esta operación sobrescribirá de forma permanente los valores de celda en su libro de origen abierto.

Se requiere el archivo de proyecto

Esta función de descifrado requiere el archivo de proyecto original de XLS Padlock (.xplp) que se utilizó para crear la aplicación. XLS Padlock usa la configuración de su proyecto (como la Secret Key) para cifrar y descifrar los archivos de guardado. Sin el archivo de proyecto correcto, no podrá descifrar los guardados. **Nunca comparta sus archivos de proyecto de XLS Padlock.**

Bloquear los guardados a un equipo

Esta opción impide que los usuarios finales compartan sus archivos de guardado. Cuando está habilitada, la aplicación crea **archivos de guardado bloqueados por hardware**, que solo se pueden abrir en el equipo donde se crearon. Si un usuario intenta abrir un archivo de guardado en un equipo diferente, no se podrá cargar.

La aplicación incrusta el **ID de sistema** único del equipo en el archivo de guardado y lo verifica al cargarlo.

CONSEJO

Como propietario de la aplicación, usted todavía puede descifrar por sí mismo cualquier archivo de guardado bloqueado por hardware mediante la función [Decrypt Save File](#) de XLS Padlock.

Advertencia

Esta función solo está disponible si ha seleccionado el modo de guardado completo (Full Save mode).

Referencias externas e hipervínculos

XLS Padlock protege un libro por archivo EXE. Si su libro utiliza referencias externas o requiere archivos adicionales, debe actualizar sus rutas para garantizar que funcionen correctamente después de la compilación.

XLS Padlock ofrece dos formas principales de gestionar los archivos externos:

- Añadirlos como [Companion Files](#).
- Utilizar rutas dinámicas en hipervínculos o código VBA.

Para gestionar rutas dinámicas, XLS Padlock ofrece una función llamada `PLEvalVar` que puede utilizarse directamente en fórmulas de Excel o invocarse desde VBA.

Esta función toma un argumento de cadena:

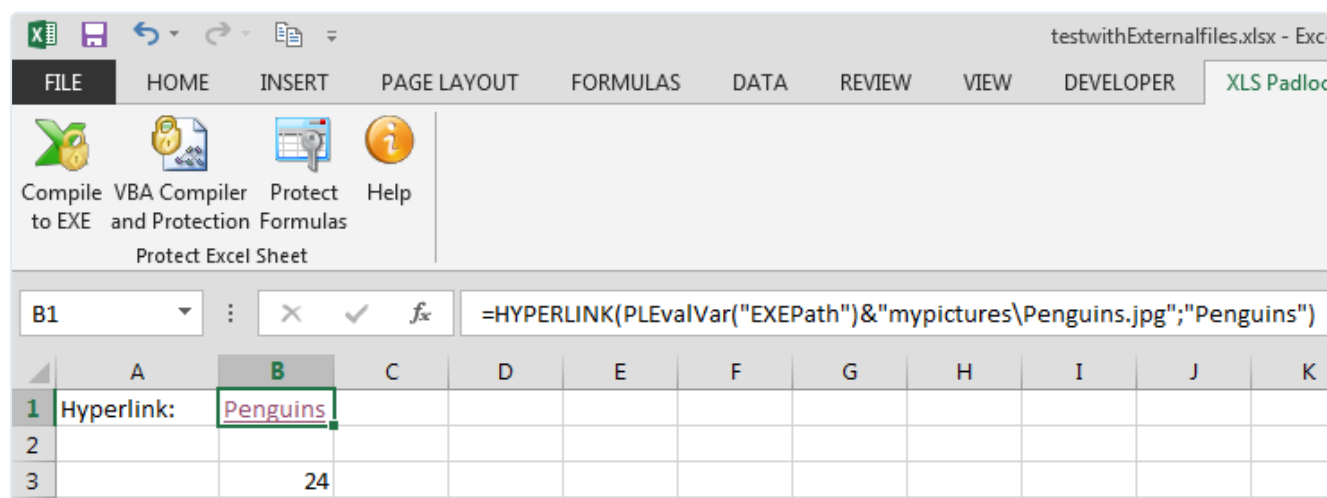
- `=PLEvalVar("EXEPath")` devuelve la ruta completa a la carpeta que contiene el archivo EXE de la aplicación (con la barra invertida final).
- `=PLEvalVar("XLSPath")` devuelve la ruta completa a la carpeta que contiene el libro compilado en tiempo de ejecución (con la barra invertida final).

NOTA

Esta carpeta es una carpeta virtual y, por tanto, no puede colocar archivos reales en ella. Solo resulta útil si trabaja con Companion files (consulte [Add Companion Files](#)).

Ejemplo 1

Tiene hipervínculos a archivos de imagen externos. Estos archivos de imagen están en la misma carpeta que el archivo XLS del libro (o en una subcarpeta).



Tiene un hipervínculo en una celda definido por:

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

Para que funcione con XLS Padlock, debe copiar todos los archivos de imagen externos en la misma carpeta que el archivo EXE generado con XLS Padlock. A continuación, tiene que modificar todos los hipervínculos para insertar la función `PLEvalVar("EXEPath")` que devuelve la ruta a esa carpeta.

En nuestro caso, esto se convertirá en:

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

Advertencia

Los archivos externos deben implementarse en la misma carpeta que el archivo EXE final de la aplicación. También es una buena práctica evitar el uso de espacios en los nombres de archivo.

Esto también funciona para archivos en subcarpetas. Un enlace como:

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

...debe cambiarse a:

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

Ejemplo 2

Para acceder a archivos externos con VBA, puede utilizar una función auxiliar que construye la ruta completa a un archivo ubicado en la misma carpeta que el EXE.

👉 Para obtener una explicación detallada y un fragmento de código reutilizable, consulte la guía sobre cómo obtener la ruta a un archivo en la carpeta del 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
```

Obtener la ruta junto al libro compilado

Esta función VBA recupera la ruta completa a un archivo ubicado en el mismo directorio que el EXE de su aplicación. Resulta especialmente útil para acceder a recursos externos o Companion files que distribuye junto con su libro protegido.

👉 Inserte la siguiente función en un módulo 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
```

A continuación, puede invocar la función:

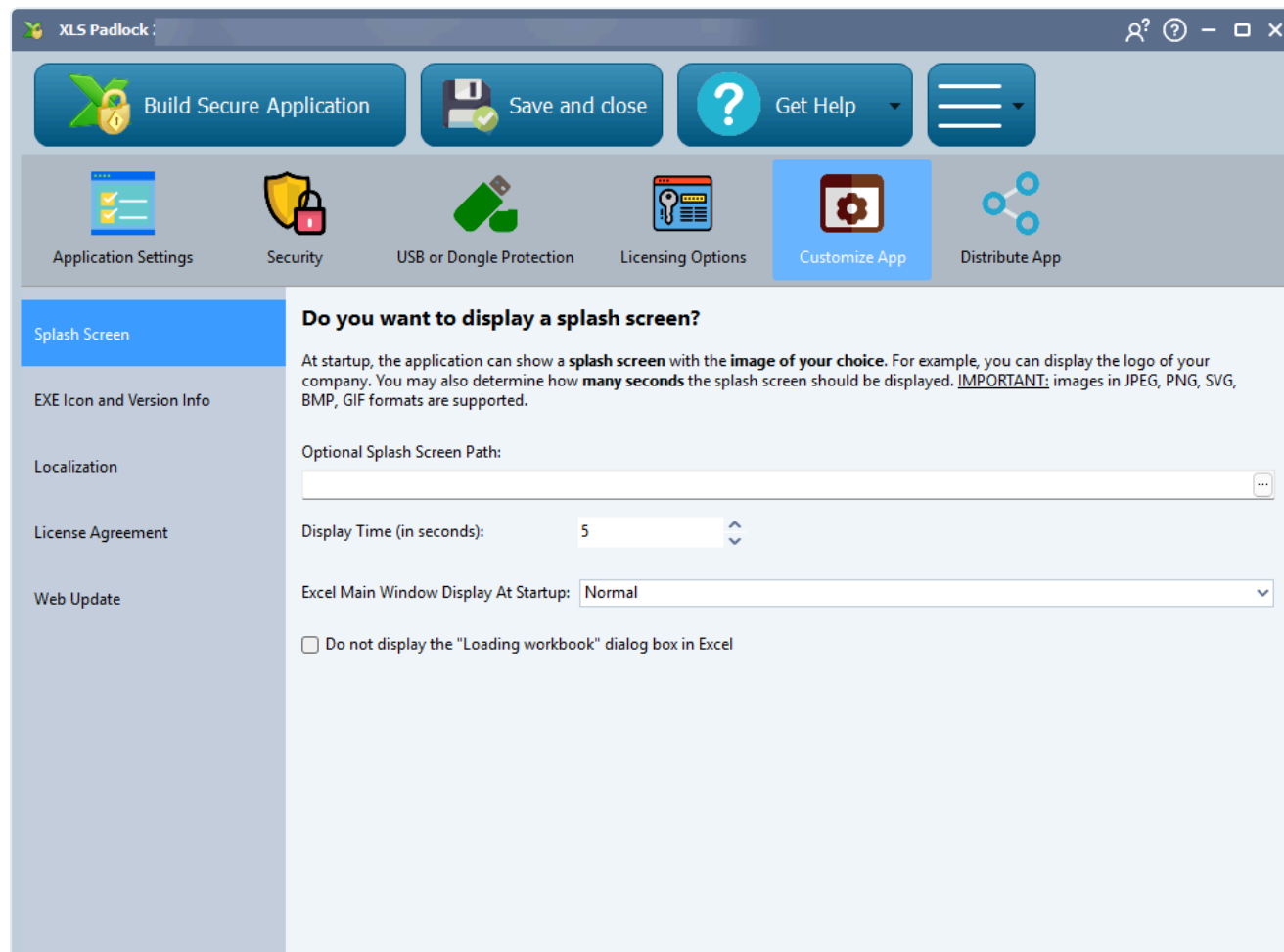
```
Sub Test_File()
    DoSomethingWith(PathToFile("data.xls"))
End Sub
```

👉 Consulte también: [VBA API Cookbook & Recipes](#)

Personalizar la aplicación

XLS Padlock le ofrece varias formas de **personalizar su aplicación de Excel compilada**, lo que le permite adaptar su apariencia, sus mensajes y su comportamiento para los usuarios finales.

Este tema proporciona un resumen de las opciones disponibles en "**Customize App**" en XLS Padlock:



Splash Screen

Muestre una **splash screen** (pantalla de bienvenida) personalizada cuando se inicia su aplicación, por ejemplo, el logotipo de su empresa o una imagen de bienvenida con su marca.

Puede elegir el formato de imagen (JPEG, PNG, BMP, SVG o GIF), establecer cuántos segundos permanece visible y controlar cómo se comporta la ventana de Excel durante el inicio.

[➔ Más información sobre la configuración de Splash Screen](#)

EXE Icon and Version Info

Sustituya el icono predeterminado de XLS Padlock por el **icono de su propia aplicación**.

También puede especificar información de versión, como el **product name**, **company name**, **copyright** y **file description**, todo ello visible en el cuadro de diálogo de propiedades de archivo de Windows.

Esto le ayuda a dar a su EXE la imagen de una aplicación independiente y profesional.

→ [Más información sobre EXE Icon y Version Info](#)

Localization

Personalice o traduzca todos los **cuadros de diálogo, mensajes y avisos integrados** que muestra su libro protegido.

XLS Padlock ofrece un sistema de localización flexible para que pueda adaptar su aplicación a diferentes idiomas o crear mensajes de usuario totalmente personalizados.

→ [Más información sobre Localization](#)

License Agreement

Añada un **license agreement** (contrato de licencia) que los usuarios deben aceptar antes de ejecutar por primera vez su aplicación de Excel compilada.

Puede incluir texto sin formato o cargar un archivo RTF/HTML externo.

Si los usuarios rechazan el contrato, la aplicación no continuará abriéndose.

→ [Más información sobre License Agreement](#)

Web Update

Habilite las **actualizaciones a través de la web** para su aplicación de Excel.

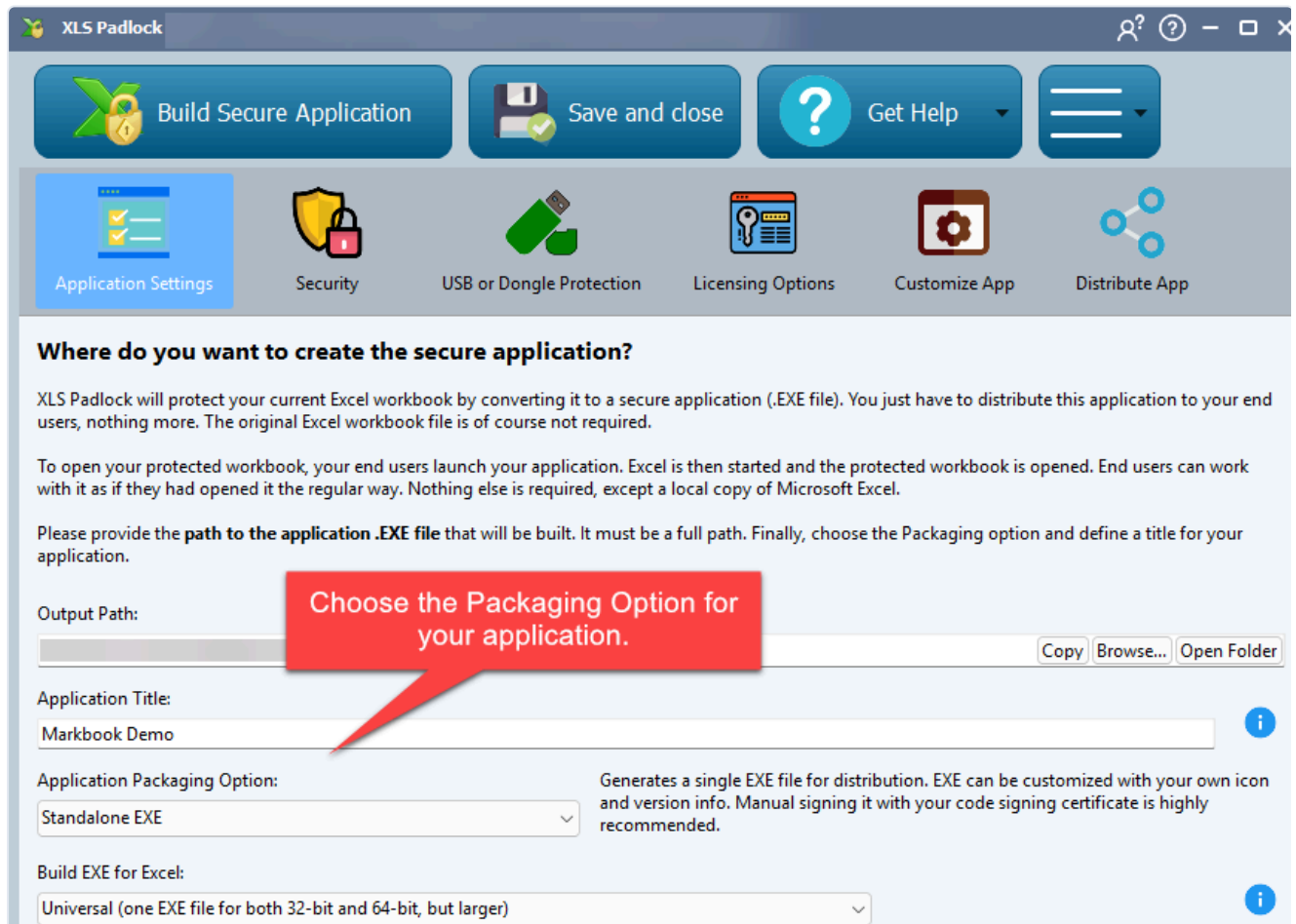
Esta función le permite notificar a los usuarios cuando hay disponible una nueva versión de su EXE y descargarla automáticamente desde su sitio web o servidor.

Es una forma eficaz de mantener a sus usuarios al día con un esfuerzo mínimo.

→ [Más información sobre Web Update](#)

Empaquetado de la aplicación

La opción **Application Packaging** determina cómo se compila su libro protegido y ofrece dos formatos distintos.



Recomendación

Recomendamos la opción **Standalone EXE** si dispone de un [certificado de firma digital](#). Aunque no es obligatorio, firmar su EXE es la mejor forma de evitar las advertencias de "Unknown App" de Windows SmartScreen y de reducir los falsos positivos del software antivirus.

Standalone EXE

Esta opción compila su libro de Excel en un único archivo ejecutable (`.exe`) independiente. Puede distribuir este único archivo a sus usuarios, que no necesitarán el libro de Excel original.

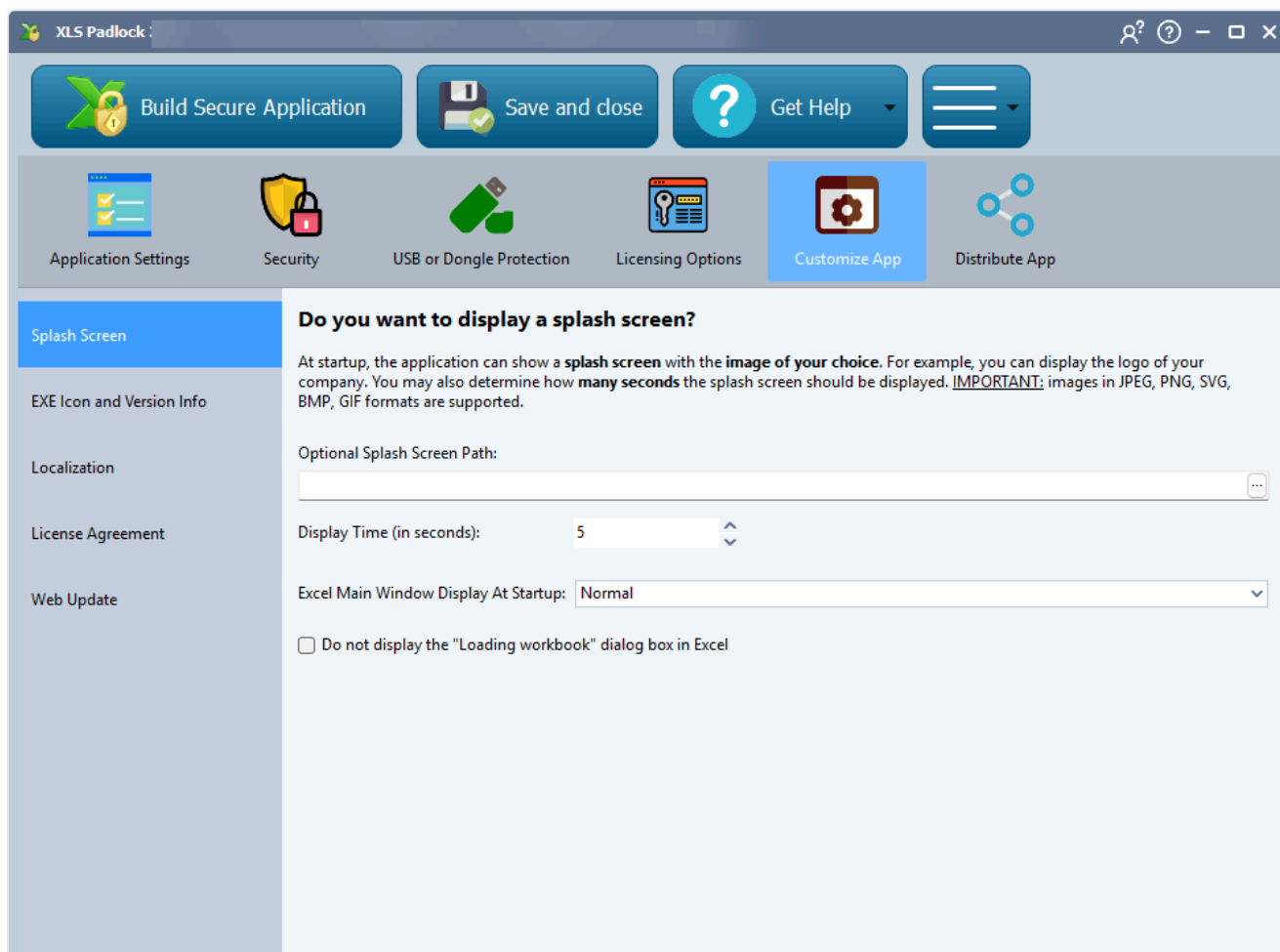
EXE + XPLAPP Application Bundle

Esta opción crea un archivo EXE junto con un archivo complementario `.bin64` y un archivo de datos `.xplapp`. El archivo EXE principal viene prefirmado por nuestra empresa, lo que ayuda a que goce de mayor confianza por parte de Windows SmartScreen y de las soluciones antivirus.

- Este método de distribución requiere que los archivos EXE, `.bin64` y `.xplapp` se mantengan todos en la misma carpeta.
- Para mayor comodidad, XLS Padlock puede agrupar estos archivos en un [único archivo Zip o en un instalador](#).
- El uso de este método reduce considerablemente las probabilidades de encontrar alertas de "Unknown Application" y falsos positivos de los antivirus.
- El archivo complementario `.bin64` garantiza la compatibilidad con las versiones de 64 bits de Excel.

Pantalla de bienvenida

Puede mostrar una **pantalla de bienvenida** (splash screen), una imagen que aparece brevemente al inicio, mientras su aplicación se está inicializando. Es una forma excelente de mostrar el logotipo de su empresa o de dar identidad de marca a su aplicación.



Configuración de la pantalla de bienvenida

Puede utilizar cualquier imagen en formato JPEG, PNG, BMP, SVG o GIF.

Puede establecer la duración durante la cual se muestra la pantalla de bienvenida. De forma predeterminada, los usuarios pueden cerrarla de inmediato haciendo clic sobre ella. Este comportamiento se puede desactivar en las [Advanced Options](#).

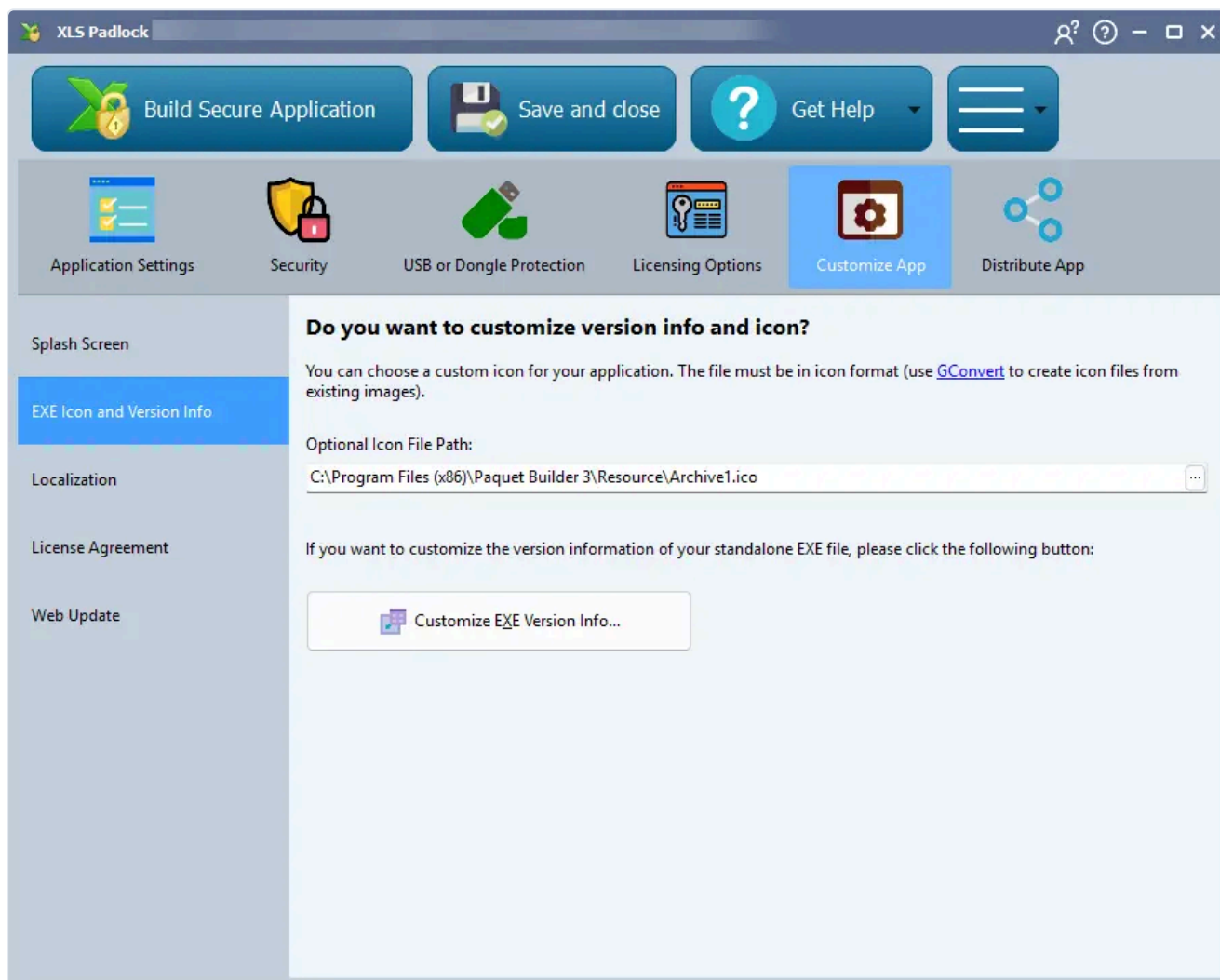
Use PNG transparentes para un aspecto único

XLS Padlock admite pantallas de bienvenida no rectangulares y semitransparentes. Use un archivo PNG de 32 bits con canal alfa para dotar a su aplicación de una apariencia moderna y personalizada.

👉 Consulte también: [No mostrar el cuadro de diálogo "Loading workbook" en Excel](#chapter-do-not-display-the-loading-workbook-dialog-box).

Cambiar el icono del EXE

XLS Padlock le permite sustituir el **icono predeterminado de la aplicación** por uno propio. Para personalizar la apariencia de su aplicación, especifique la **ruta a un archivo de icono personalizado**, que debe tener la extensión `.ico` :



XLS Padlock admite formatos de icono estándar, incluidos diversos tamaños (por ejemplo, 32x32, 48x48) y profundidades de color. Si necesita una herramienta para crear o extraer iconos, le recomendamos nuestra utilidad complementaria, [GConvert](#), que puede convertir imágenes en iconos con facilidad.

Advertencia

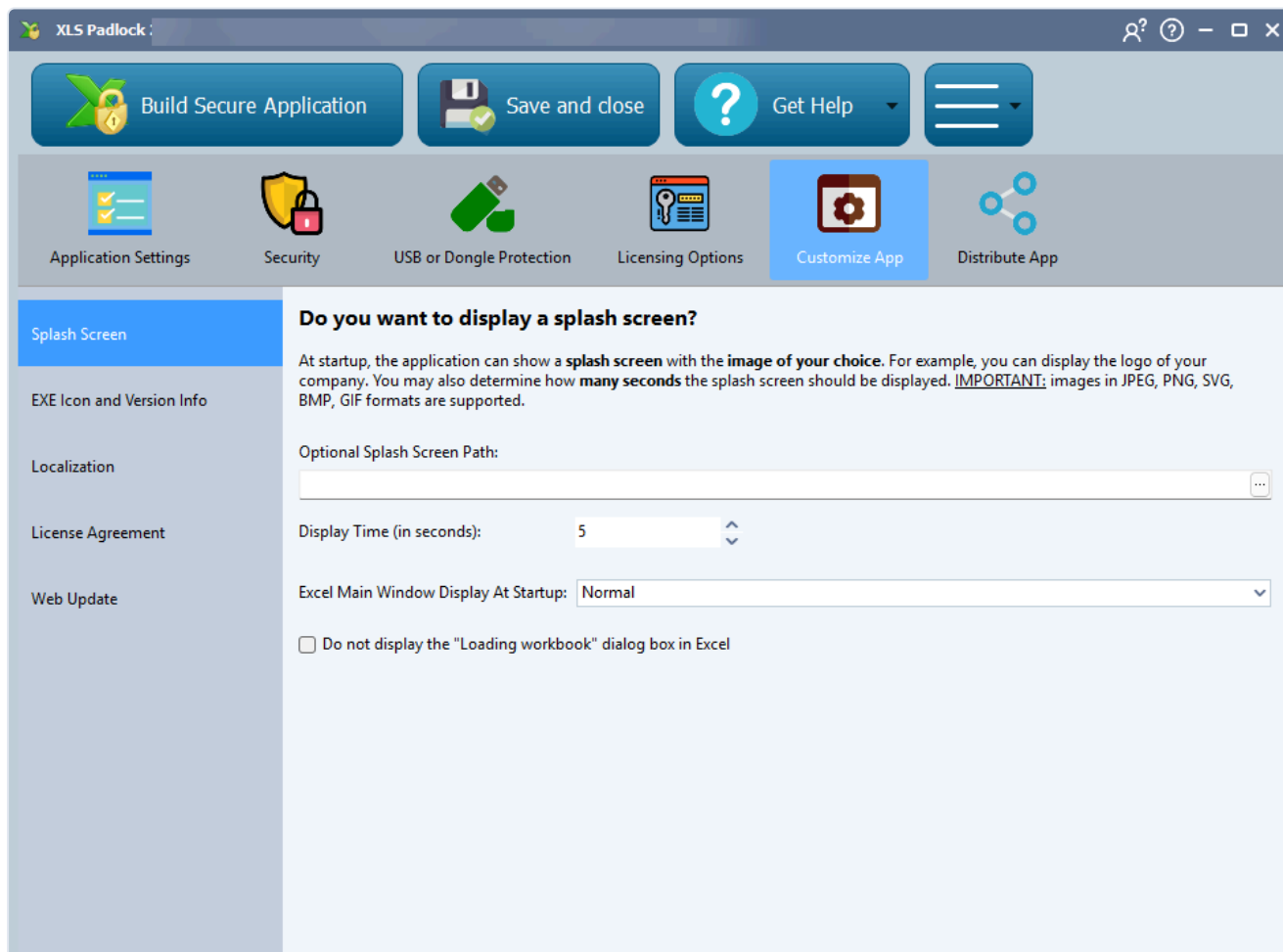
El archivo de icono debe estar disponible como archivo externo en su equipo durante el proceso de compilación.

Información

XLS Padlock también aplicará este icono al archivo EXE de la aplicación independiente, garantizando una identidad de marca coherente.

Ventana de Excel al inicio

La opción **Excel Main Window Display at Startup** le permite definir cómo se ajustará el tamaño de la ventana de Excel cuando se inicie su aplicación. Este ajuste se encuentra en la página [Splash Screen](#).



Puede elegir entre tres opciones:

- **Normal:** la ventana de Excel se abre con su tamaño y posición predeterminados. Este es el comportamiento por defecto.
- **Minimized:** la aplicación se inicia minimizada en la barra de tareas de Windows. Esto resulta útil si su aplicación se ejecuta en segundo plano o si desea mostrar un formulario personalizado antes de mostrar la ventana principal de Excel.
- **Maximized:** la ventana de Excel se abre maximizada para ocupar toda la pantalla.

¿Necesita ocultar por completo la ventana de Excel?

Si su aplicación funciona únicamente mediante código VBA y UserForms y desea que la ventana de Excel, junto con su icono de la barra de tareas, no aparezca nunca, consulte [Ejecutar como una aplicación solo de VBA \(ventana principal de Excel totalmente oculta\)](#). Cuando esa opción está activada, la elección anterior queda anulada y Excel siempre se inicia oculto.

Véase también

- [Cómo configurar la Splash Screen](#)
- [Cómo ocultar el cuadro de diálogo "Loading workbook"](#)
- [Ejecutar como una aplicación solo de VBA \(ventana principal de Excel totalmente oculta\)](#)

Ejecutar como aplicación solo VBA

Si su libro protegido funciona enteramente mediante **código VBA** y **UserForms**, y la hoja de cálculo en sí no debe ser vista nunca por sus usuarios finales, puede distribuirlo como una *aplicación solo de VBA*. La ventana principal de Excel permanece oculta de principio a fin, incluido su icono de la barra de tareas, y solo aparece en pantalla su UserForm. Su aplicación luce y se comporta como un programa independiente de Windows, sin ningún rastro visible de Excel.

Esta opción está disponible desde **XLS Padlock 2026.0**.

Cuándo utilizarla

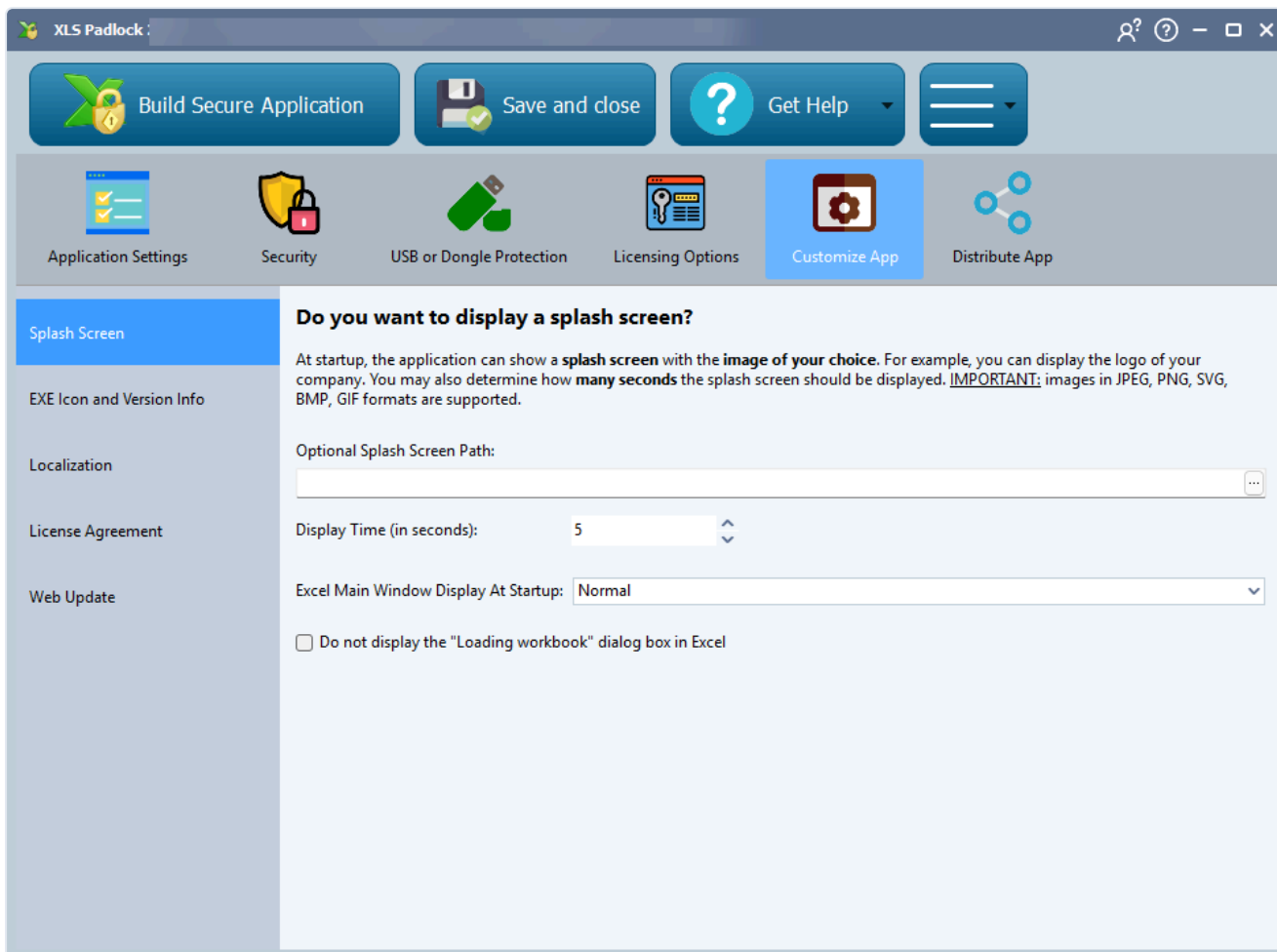
Utilice el modo solo de VBA cuando se cumplan **todas** las condiciones siguientes:

- La interfaz de usuario de su aplicación está construida con **UserForms**, no con hojas de cálculo.
- No desea que los usuarios finales vean, editen o interactúen con la hoja de cálculo subyacente.
- Desea un aspecto limpio y con su marca al inicio, sin pantalla de presentación de Excel, sin entrada de Excel en la barra de tareas.

Si su aplicación depende de hojas de cálculo visibles (cuadrículas de entrada de datos, paneles representados como celdas, etc.), **no** active esta opción, ya que las hojas de cálculo no serán visibles para sus usuarios.

Cómo activarla

1. Abra su proyecto en XLS Padlock.
2. Vaya a la página **Splash Screen** (en Application Customization).
3. Marque la opción **"Run as a VBA-only app (Excel main window fully hidden)"**.



Eso es todo por parte de XLS Padlock. Ahora debe añadir un pequeño fragmento de código VBA a su libro para que su UserForm se muestre al iniciarse la aplicación.

Añadir el código VBA de inicio

En el editor de VBA de Excel, abra el módulo `ThisWorkbook` y pegue:

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

Sustituya `UserForm1` por el nombre del formulario que desea mostrar en primer lugar.

¿Por qué ambos, la casilla *y* el código VBA?

La casilla de XLS Padlock garantiza que la ventana principal de Excel nunca se haga visible **antes** de que se ejecute su VBA (sin parpadeo, sin icono en la barra de tareas). La línea

```
Application.Visible = False en Workbook_Open
```

mantiene Excel oculto **después** de que Excel termine de inicializarse, ya que Excel restablece su visibilidad internamente durante el arranque, por lo que VBA necesita imponer la regla una vez que el libro está completamente cargado. Usar ambas capas le proporciona una experiencia de usuario limpia y sin parpadeos.

Lo que ve el usuario final

Con la casilla activada y el fragmento de VBA anterior:

- **Sin pantalla de presentación de Excel** al iniciar.
- **Sin ventana principal de Excel** en ningún momento durante la vida de la aplicación.
- **Sin entrada de "Excel" en la barra de tareas de Windows**, solo su UserForm.
- Su aplicación se comporta como un programa independiente de Windows construido en torno a su formulario.

Si también ha configurado una [splash screen](#) y ocultado el [cuadro de diálogo Loading workbook](#), toda la secuencia de inicio permanece dentro de su interfaz con su marca desde el primer fotograma.

Cerrar la aplicación

Dado que Excel está oculto, el botón estándar de cierre de Excel no está disponible para sus usuarios. Normalmente, la aplicación se cierra desde su UserForm mediante:

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

O, si desea dejar Excel en ejecución pero descargar su interfaz:

```
Unload Me
```

Interacción con otros ajustes

Ajuste	Comportamiento cuando "VBA-only app" está activado
Excel Main Window Display at Startup (Normal / Maximized / Minimized)	Anulado, Excel siempre se inicia oculto.

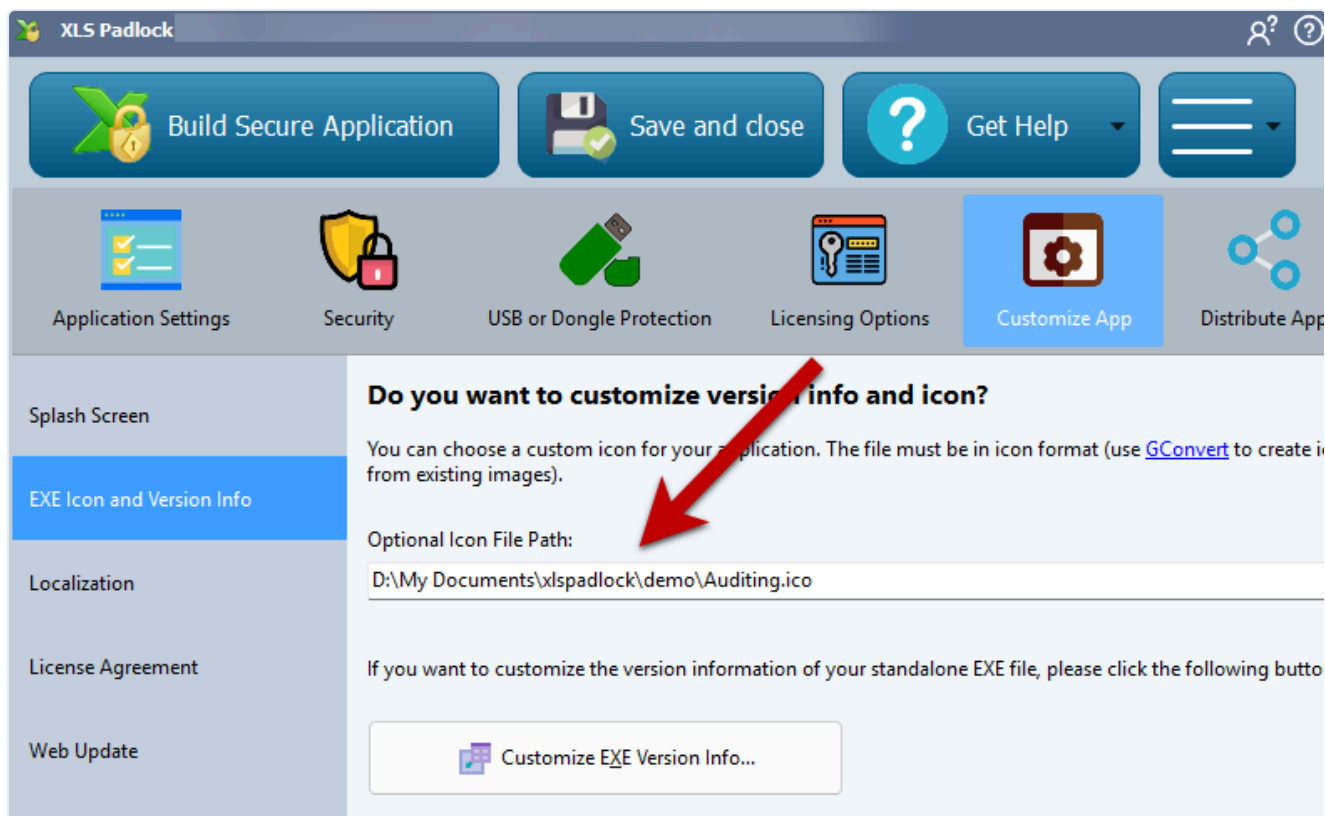
Ajuste	Comportamiento cuando "VBA-only app" está activado
Do not display the "Loading workbook" dialog box	Recomendable activarlo también, para un inicio totalmente silencioso.
Splash Screen	Funciona según lo configurado. Se muestra incluso antes de que Excel se cargue.

Véase también

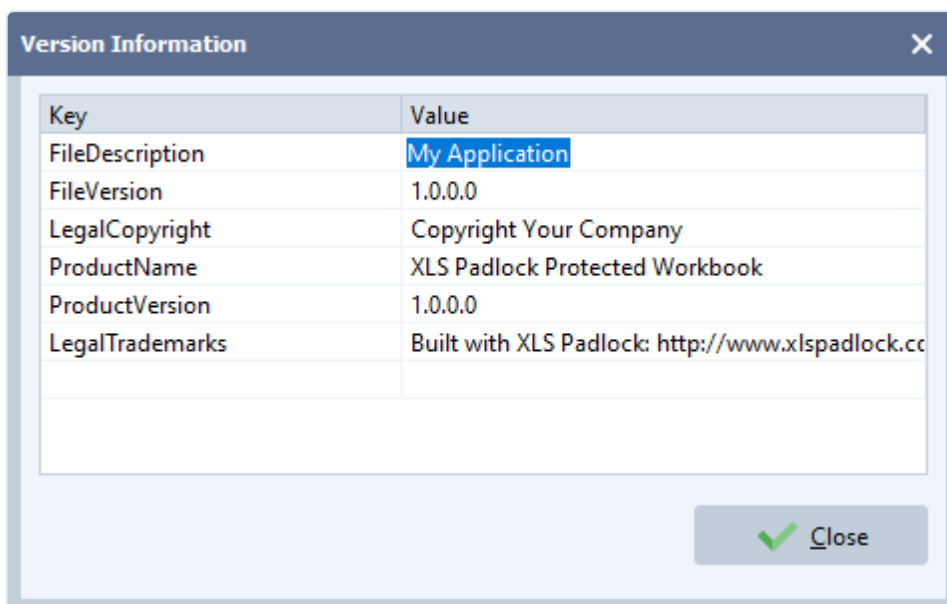
- [Cómo configurar la Splash Screen](#)
- [Cómo definir la presentación de la ventana principal de Excel al inicio](#)
- [Cómo ocultar el cuadro de diálogo "Loading workbook"](#)
- [Recetario de la API de VBA](#)

Información de versión del EXE

La **información de versión** de un archivo ejecutable es una sección de recursos especial que contiene datos como el número de versión del archivo, el sistema operativo previsto, el nombre de archivo original y la información de copyright. Estos datos se incrustan en el EXE compilado. Cuando se incluyen, los usuarios pueden ver esta información haciendo clic con el botón derecho en el icono del programa, seleccionando "Properties" y navegando a la pestaña "Details" (o pulsando `Alt+Enter` sobre el archivo en el Explorador).



XLS Padlock le permite incrustar su propia información de versión en el archivo EXE independiente. Haga clic en "**Customize EXE Version Info**" para abrir una ventana con los siguientes campos:



- **File Description:** una breve descripción del contenido de su aplicación.
- **Company Name:** el nombre de su empresa.
- **File Version:** el número de versión de su archivo `.exe`, con el formato `X.X.X.X` (por ejemplo, `1.20.34.45`). Este valor también lo utiliza la [función Web Update](#) para comprobar si hay nuevas versiones de la aplicación.
- **Legal Copyright:** su aviso de copyright, como "Copyright © [año] Your Company. All rights reserved."
- **Product Name:** el nombre de su producto o aplicación, que generalmente coincide con el título de su aplicación.
- **Product Version:** el número de versión de su producto. A menudo coincide con la File Version.
- **Legal Trademarks:** cualquier marca comercial legal que desee incluir.

Información

La File Version la utiliza la [función Web Update](#).

Parámetros de línea de comandos

El archivo EXE de su aplicación compilada admite varios **modificadores de línea de comandos** (también conocidos como argumentos) que automatizan acciones concretas al iniciarse. Puede usarlos en accesos directos o scripts para controlar cómo se comporta la aplicación cuando arranca.

CONSEJO

👉 Estos argumentos de línea de comandos también pueden recuperarse en tiempo de ejecución mediante la [API de VBA para leer parámetros de línea de comandos](#).

Modificadores disponibles

-deact

Inicia el proceso de desactivación, lo que permite a un usuario trasladar su licencia a otro equipo.

```
MYAPP.EXE -deact
```

-deL

Borra todos los archivos de guardado seguro de la aplicación y carga el libro original. **Úselo con precaución, ya que esta acción no se puede deshacer.**

```
MYAPP.EXE -deL
```

-enterkey

Abre el cuadro de diálogo de activación, lo que permite al usuario introducir una nueva clave de activación. Resulta útil para reemplazar una clave antigua o caducada.

```
MYAPP.EXE -enterkey
```

-load

Abre un cuadro de diálogo de archivos que solicita al usuario que seleccione un archivo de guardado para cargar.

```
MYAPP.EXE -load
```

-reset

Descarta cualquier cambio sin guardar de la última sesión y carga el libro original sin modificar. Esto no elimina ningún archivo de guardado `.xlsc`.

```
MYAPP.EXE -reset
```

-webupdate

Obliga a la aplicación a comprobar inmediatamente si hay una nueva versión en línea, según sus ajustes de "Web Update".

```
MYAPP.EXE -webupdate
```

Cargar un archivo de guardado concreto

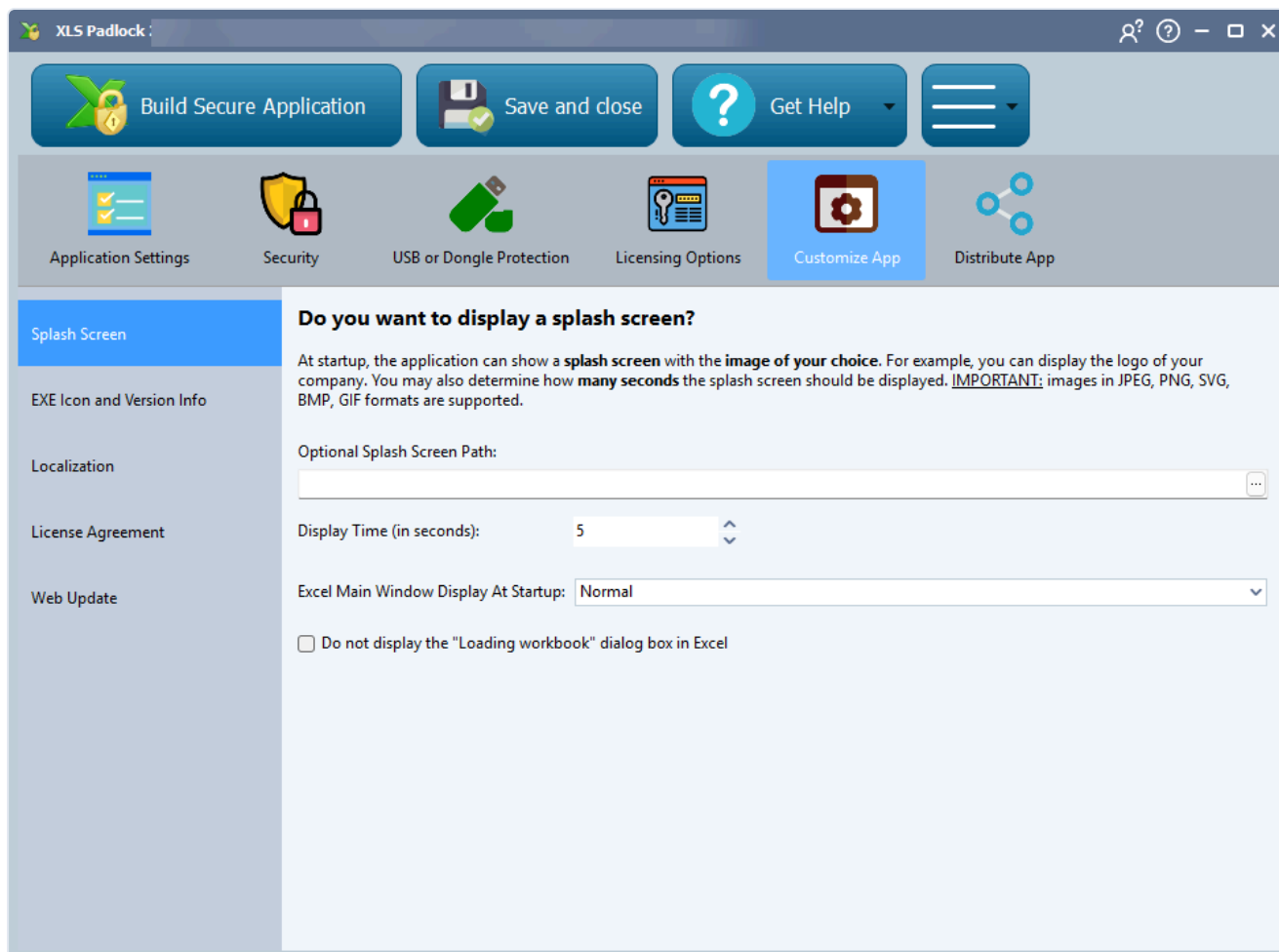
También puede iniciar la aplicación con la ruta completa a un archivo de guardado seguro (`.xlsc` o `.xlsce`) como argumento. Esto cargará automáticamente el archivo especificado al iniciarse, sin ninguna solicitud.

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

Ocultar el cuadro de diálogo de carga

De forma predeterminada, cuando se inicia su aplicación protegida, Excel muestra un pequeño cuadro de diálogo con el mensaje **"Loading workbook, please wait..."**.

Puede ocultar este cuadro de diálogo activando la opción **"Do not display the "Loading workbook" dialog box in Excel"**, situada en la página [Splash Screen](#). Resulta útil para crear una experiencia de inicio más limpia, especialmente cuando ya está utilizando una pantalla de bienvenida personalizada.



También es posible [ocultar este mensaje de forma anticipada mediante código VBA](#), lo que le da más control sobre cuándo desaparece el cuadro de diálogo.

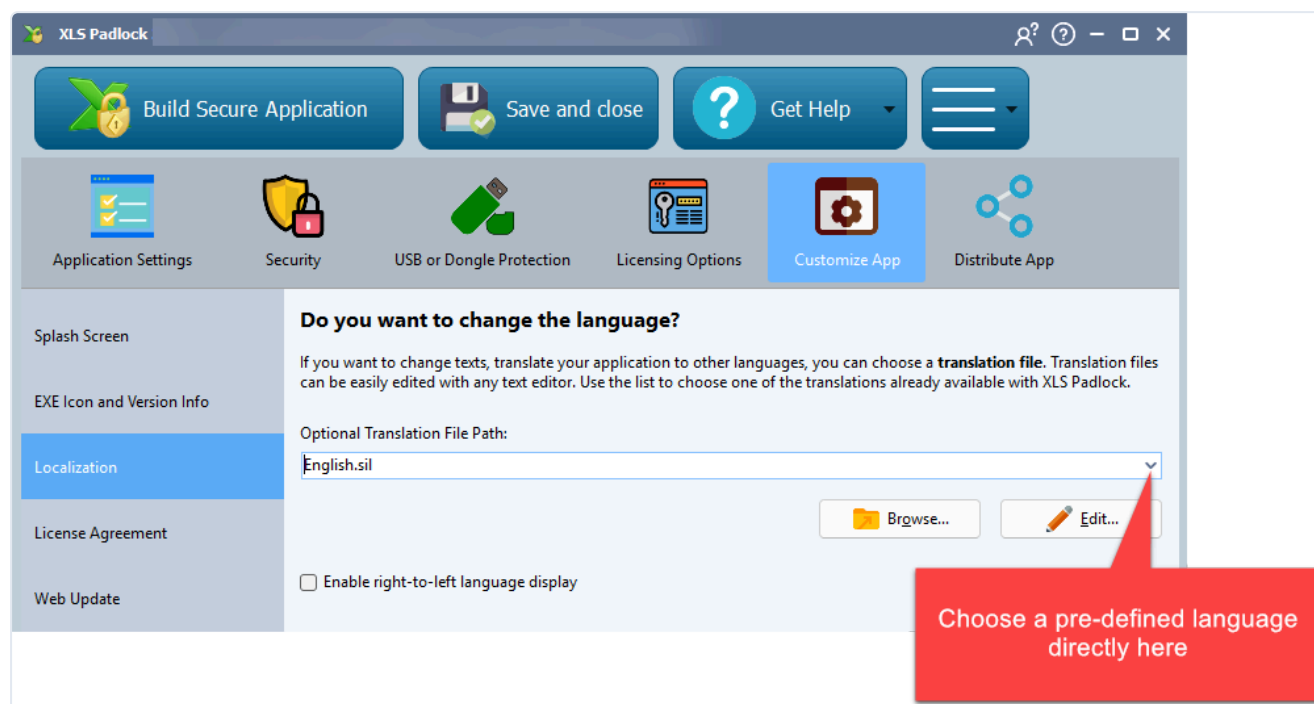
👉 Véase también

- [Cómo configurar la Splash Screen](#)
- [Cómo definir la visualización de la ventana principal de Excel al iniciarse](#)

Localización y traducción

XLS Padlock proporciona **soporte de localización** para sus libros protegidos, lo que le permite mostrar todo el texto de la interfaz (títulos de diálogo, mensajes, solicitudes y advertencias en tiempo de ejecución) en el idioma que elija.

Puede configurar la localización para la aplicación de su libro protegido aquí:



Formato del archivo de traducción

Desde XLS Padlock 2026, los archivos de traducción usan el formato **GNU gettext estándar** con la extensión `.po` (y su contraparte binaria compilada `.mo`). Es el mismo formato que utilizan las distribuciones de Linux, Python, PHP, WordPress y muchísimas otras aplicaciones, lo que significa que puede editar las traducciones con cualquiera de las decenas de herramientas gettext maduras disponibles, incluido el editor gratuito y multiplataforma [Poedit](#).

La aplicación compilada integra un único recurso `.mo` en el momento de la compilación, cargado por completo desde la memoria en tiempo de ejecución: no se extrae ningún archivo temporal al disco.

XLS Padlock incluye archivos `.po` ya traducidos para **10 idiomas**, disponibles en la subcarpeta `locale\` del directorio de instalación de XLS Padlock:

Archivo	Idioma
en.po	Inglés
fr.po	Francés
es.po	Español
pt.po	Portugués (Brasil)

Archivo	Idioma
nL.po	Neerlandés
de.po	Alemán
ar.po	Árabe (compatible con diseño de derecha a izquierda)
it.po	Italiano
zh_CN.po	Chino simplificado
zh_TW.po	Chino tradicional

Solicitar un nuevo idioma

Si necesita un idioma que no figura en la lista, [póngase en contacto con nosotros](#). También puede crear su propio archivo `.po` a partir de cualquiera de los anteriores como punto de partida: consulte *Edición de archivos de traducción* más abajo.

Cómo definir un idioma para su libro protegido

- En la página **Localization**, haga clic en **Browse** junto al campo *Translation file*. El cuadro de diálogo se abre de forma predeterminada en la carpeta `locale\` de su instalación de XLS Padlock, donde se encuentran los 10 archivos `.po` ya preparados. Elija el idioma que desee o navegue hasta un archivo `.po` personalizado propio.
- También puede pegar la ruta completa manualmente en el campo *Translation file*, o dejarlo vacío para usar la traducción inglesa predeterminada integrada en XLS Padlock.

IMPORTANTE

El archivo de traducción seleccionado debe ser accesible en su ruta especificada **cuando compile su aplicación**. XLS Padlock lee y compila este archivo durante el proceso de compilación e integra el recurso `.mo` resultante directamente en el EXE protegido: no hay ninguna dependencia en tiempo de ejecución del archivo `.po` original.

Edición de archivos de traducción

Los archivos `.po` son archivos de texto UTF-8 sin formato. Puede editarlos de dos maneras:

- **Con Poedit** (poedit.net, gratuito, multiplataforma): recomendado. Muestra las cadenas de origen y traducidas una al lado de la otra, marca las entradas dudosas (fuzzy) y sin traducir, valida la sintaxis al guardar y produce una salida UTF-8 limpia.
- **Con cualquier editor de texto** (Notepad++, VS Code, ...). Guárdelo como **UTF-8 sin BOM** y mantenga un par `msgid / msgstr` por bloque. Las líneas que empiezan por `#` son comentarios y no se muestran al usuario final.

Le recomendamos que **copie uno de los archivos `.po` ya preparados** a una nueva ubicación (junto a su libro, por ejemplo), modifique la copia y, a continuación, indique a XLS Padlock su nuevo archivo.

Modificar archivos dentro de la carpeta de instalación de XLS Padlock generalmente requiere derechos de administrador y una futura actualización podría sobrescribirlos.

Migración desde los archivos de traducción `.sil` heredados

Si personalizó la interfaz de usuario de su libro protegido en XLS Padlock 2025.3 o anterior, su traducción se almacenó como un archivo `.sil` (el formato de traducción heredado basado en texto que usaban las versiones anteriores). XLS Padlock 2026 ya no lee archivos `.sil` en el momento de la compilación: el formato ha sido reemplazado por `.po`.

Para evitar volver a escribir nada, la distribución de XLS Padlock incluye un pequeño **ayudante de migración SIL → PO** en `tools\sil_to_po\` dentro del árbol de fuentes:

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

Esto produce `customer_fr.po` en la carpeta actual, fusionado automáticamente con el `.po` canónico actual de modo que:

- se copia la columna que coincide con su idioma de destino,
- las cadenas añadidas desde que se creó su `.sil` aparecen con un `msgstr` vacío (listas para rellenar en Poedit),
- las cadenas cuyo origen inglés cambió ligeramente entre versiones se marcan como `#, fuzzy` para su revisión,
- las cadenas que ya no existen en XLS Padlock se añaden al final como entradas obsoletas `#~` (o se descartan por completo con `--drop-orphans`).

Para migrar varios idiomas en una sola ejecución, sepárelos con comas:

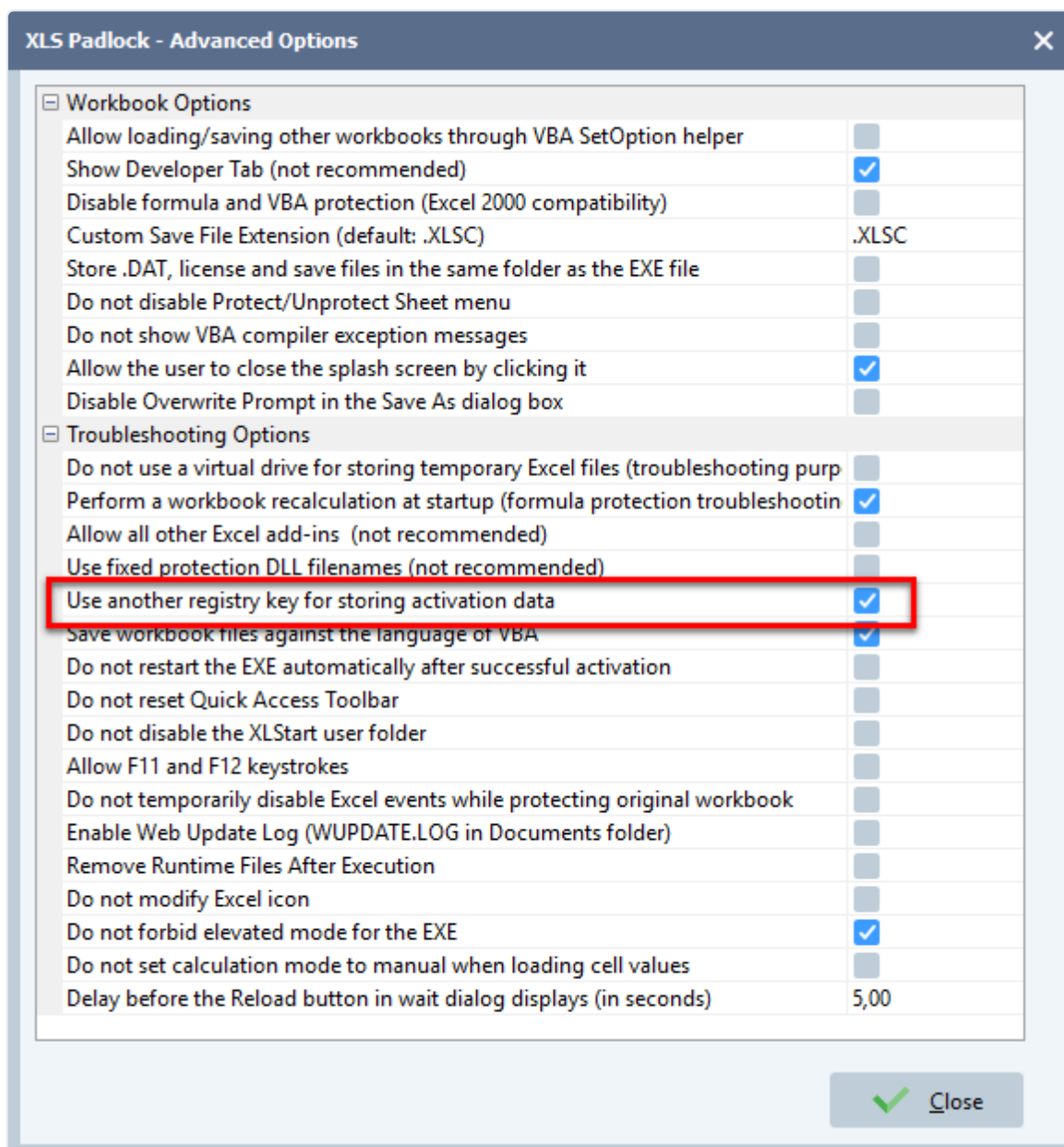
```
python tools/sil_to_po/sil2po_customer.py your_translation.sil --lang fr,es,de --stats
```

Los archivos `.po` resultantes son utilizables de inmediato desde el selector *Translation file* del XLS Padlock Designer. El script de ayuda y su `README.md` complementario (referencia completa de opciones, algoritmo de coincidencia, ejemplos resueltos) se incluyen en la subcarpeta `tools\sil_to_po\` del directorio de instalación de XLS Padlock: abra `README.md` en cualquier editor de texto para leer la documentación completa de forma local.

Se requiere Python 3.8 o posterior para ejecutar el ayudante. Sin dependencias externas de Python: el script es autónomo.

Opciones avanzadas

En “Application Settings”, haga clic en el botón “Configure Advanced Options” para mostrar la siguiente ventana:



Solo para usuarios avanzados

Estas opciones están destinadas a usuarios avanzados. Cámbielas únicamente si comprende su finalidad o si así se lo indica nuestro soporte técnico.

Opciones avanzadas disponibles

- **Allow loading/saving other workbooks through VBA SetOption helper:** omisión de seguridad para operaciones VBA específicas. Consulte [Cargar/guardar libros mediante el ayudante VBA SetOption](#).

- **Show Developer Tab:** hace visible la pestaña Developer de Excel (no recomendado por motivos de seguridad).
- **Disable formula and VBA protection:** para compatibilidad con Excel 2000.
- **Custom Save File Extension:** cambia la extensión predeterminada `.XLSC` de los archivos guardados.
- **Store .DAT, license and save files in the same folder as the EXE file:** habilita el modo portable.
- **Do not disable Protect/Unprotect Sheet menu:** mantiene habilitadas las opciones de menú de protección de hojas.
- **Do not show VBA compiler exception messages:** oculta los mensajes de error detallados del VBA Compiler. Consulte [Desactivar la información de depuración](#).
- **Allow the user to close the splash screen by clicking it:** permite a los usuarios omitir la pantalla de bienvenida.
- **Disable Overwrite Prompt in the Save As dialog box:** evita el mensaje "do you want to overwrite?".
- **Do not use a virtual drive for storing temporary Excel files:** con fines de resolución de problemas.
- **Perform a workbook recalculation at startup:** para resolver problemas de protección de fórmulas.
- **Allow all other Excel add-ins:** reduce la seguridad al habilitar todos los complementos (no recomendado).
- **Use fixed protection DLL filenames:** no recomendado por motivos de seguridad.
- **Use another registry key for storing activation data:** solución alternativa para conflictos del registro.
- **Save workbook files against the language of VBA:** resuelve algunos problemas de localización.
- **Do not restart the EXE automatically after successful activation:** evita el reinicio automático.
- **Do not reset Quick Access Toolbar:** conserva las personalizaciones del usuario en la QAT.
- **Do not disable the XLStart user folder:** permite la ejecución de los complementos de la carpeta XLStart del usuario.
- **Allow F11 and F12 keystrokes:** habilita estas teclas de función.
- **Do not temporarily disable Excel events while protecting original workbook:** con fines de resolución de problemas.
- **Enable Web Update Log:** crea un archivo `WUPDATE.LOG` en la carpeta Documentos del usuario para depurar la función de actualización web.
- **Remove Runtime Files After Execution:** limpia los archivos temporales.
- **Do not modify Excel icon:** evita que XLS Padlock altere el icono de la ventana de Excel.
- **Do not forbid elevated mode for the EXE:** permite que la aplicación se ejecute como administrador.
- **Do not set calculation mode to manual when loading cell values:** con fines de resolución de problemas.
- **Delay before the Reload button in wait dialog displays (in seconds):** personaliza el diálogo de espera.

Ignorar errores de procesamiento

La opción "Ignore errors when processing the workbook (internal protection)" permite que el proceso de compilación continúe aunque se encuentren errores al abrir su libro de Excel. Esta función está diseñada para agilizar el flujo de trabajo de protección suprimiendo los mensajes de error que, de otro modo, podrían interrumpir el proceso.

Cuando esta opción está habilitada, XLS Padlock no se detendrá ni mostrará mensajes de error si encuentra problemas durante la fase de apertura del libro.

Esta opción no corrige errores

Habilitar esta opción no resuelve ningún problema subyacente del libro; simplemente los omite durante la compilación. Después de proteger su libro con esta opción habilitada, debe probarlo a fondo para asegurarse de que todas las funciones se comporten como se espera.

Si sigue encontrando errores en el libro compilado, considere habilitar la opción alternativa, "[Usar la automatización de Excel para la protección de fórmulas](#chapter-formula-protection-method)", que puede mitigar problemas que el método estándar de protección interna no resuelve.

Desactivar la información de depuración

De forma predeterminada, si el VBA Compiler detecta un error en el código compilado en tiempo de ejecución, muestra un mensaje de error detallado para ayudarle a depurar el problema.

Si no desea mostrar esta información a sus usuarios finales, puede desactivarla. Vaya a [Configurar opciones avanzadas](#) y desactive la opción: **“Do not show VBA compiler exception messages”**.

Interfaz personalizada de Excel

Como se explica en la documentación de Microsoft, puede personalizar la interfaz de usuario de Excel colocando un archivo `.officeUI` creado especialmente en una carpeta de usuario específica.

XLS Padlock le permite especificar su propio Excel.officeUI para la aplicación segura. Reemplazará automáticamente la interfaz predeterminada del usuario (si está disponible).

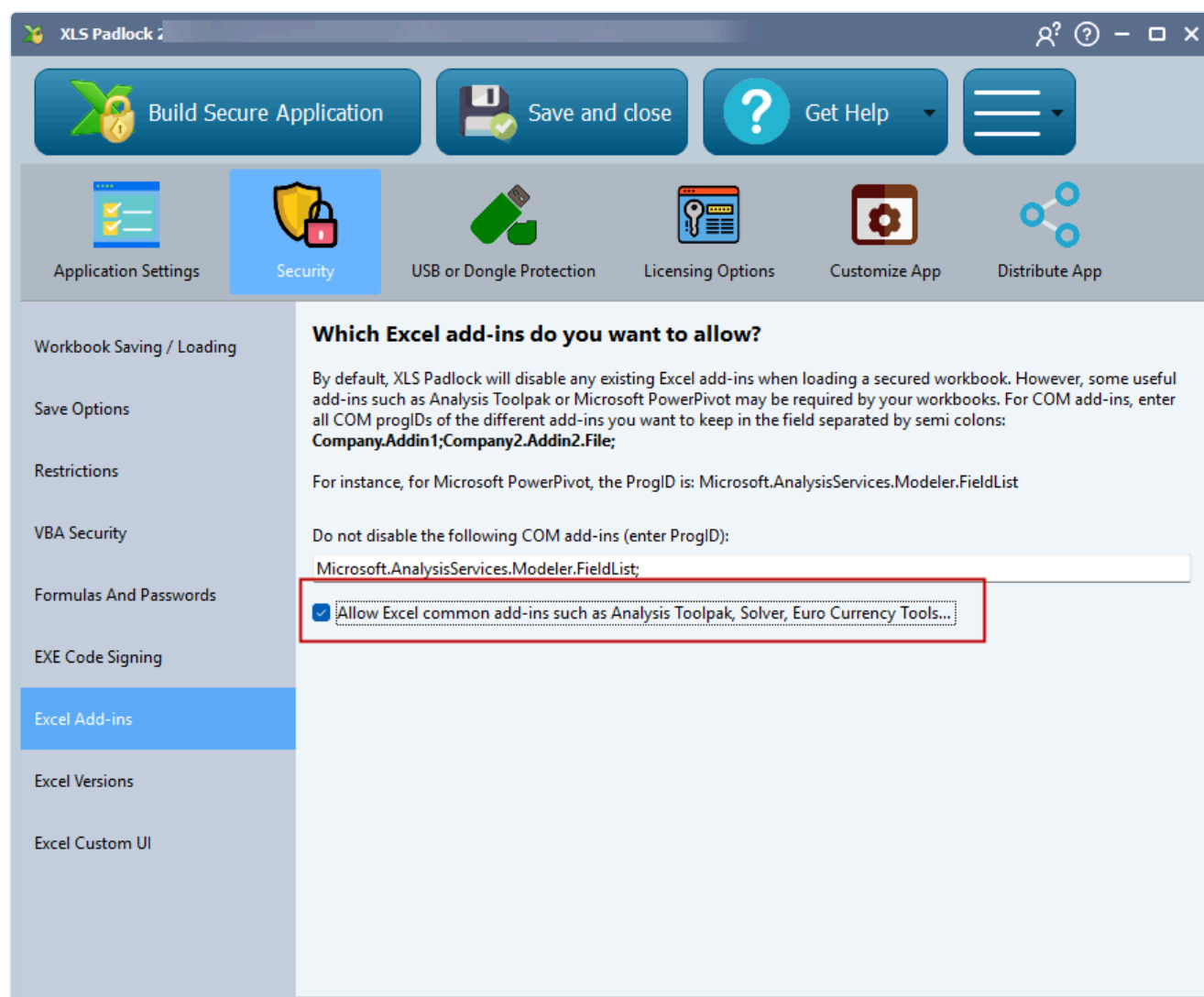
Solo necesita proporcionar la ruta completa a su archivo `.officeUI`. XLS Padlock lo compilará directamente en el EXE final, por lo que no necesita distribuir el archivo `.officeUI` por separado.

Para obtener más información sobre la creación de archivos XML de interfaz de usuario personalizada, consulte la [documentación oficial de Microsoft Office](#).

Complementos de Excel

De forma predeterminada, XLS Padlock desactiva la mayoría de los complementos de Excel cuando se carga un libro protegido. Esta es una medida de seguridad para crear un entorno controlado para su aplicación.

Sin embargo, si su libro depende de complementos específicos para funcionar correctamente, puede volver a activarlos de forma selectiva. Esta página explica cómo permitir tanto los complementos COM como los complementos integrados habituales de Excel.



Mantener activados complementos COM específicos

Si su libro requiere un complemento COM (como Microsoft PowerPivot), debe especificar su ProgID para evitar que XLS Padlock lo desactive.

En el campo "Do not disable the following COM add-ins", introduzca el ProgID de cada complemento que desee mantener activado. Si tiene más de uno, sepárelos con puntos y coma (;).

Ejemplo: Para mantener Microsoft PowerPivot activado, introduciría sus ProgIDs:

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

Encontrar un ProgID

El ProgID (identificador programático) es un nombre único almacenado en el registro de Windows para un componente COM. A menudo puede encontrarlo en la documentación del complemento o inspeccionando el registro. Microsoft proporciona más información sobre este tema en su sitio web de soporte.

Permitir complementos habituales de Excel

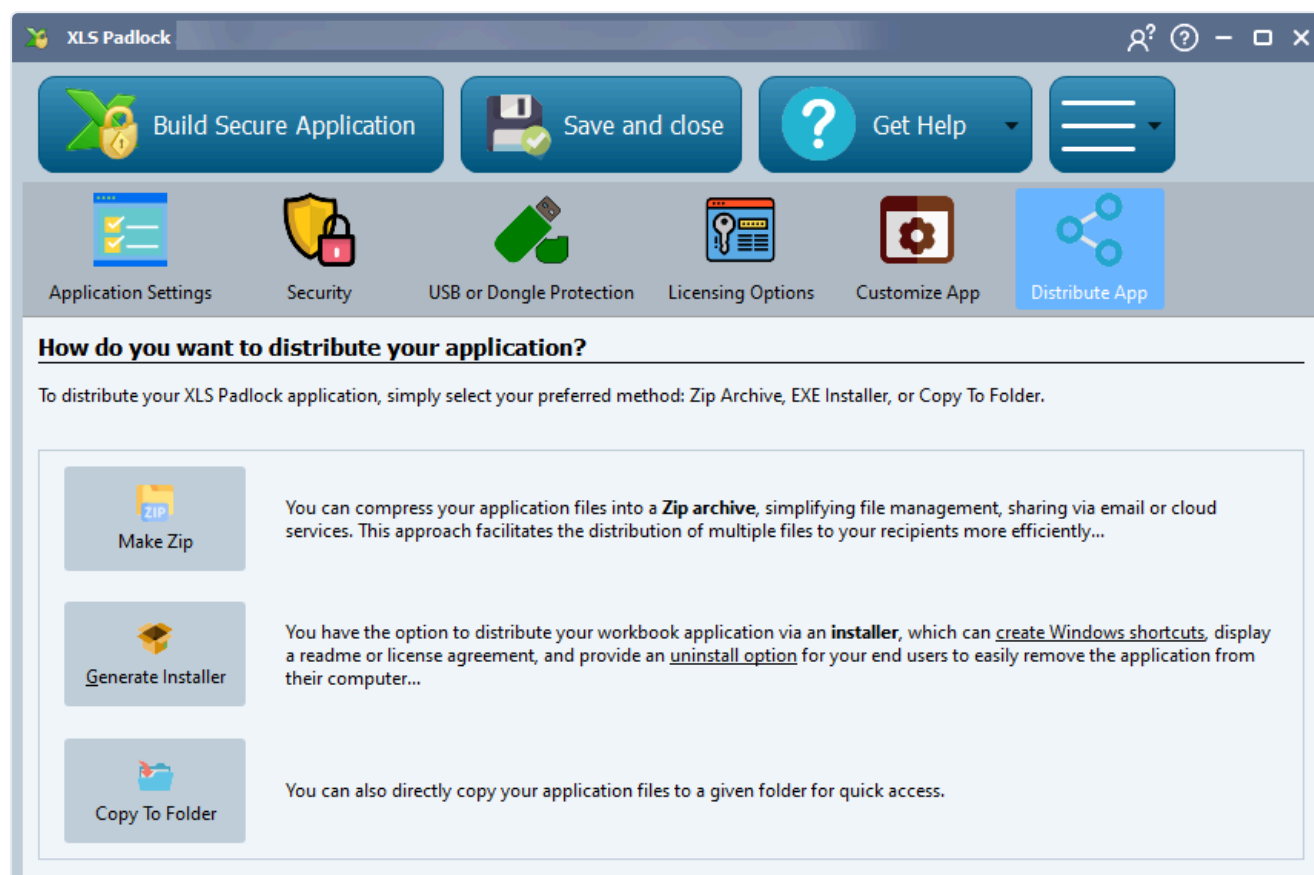
Excel también incluye varios complementos integrados populares, como **Analysis ToolPak**, **Solver** y **Euro Currency Tools**.

Si su libro utiliza funciones de cualquiera de estos, active la opción **"Allow Excel common add-ins..."**. Esto garantizará que se carguen y estén disponibles cuando se inicie su aplicación protegida.

Distribuir la aplicación

Una vez que haya [convertido su libro en un ejecutable](#) (.EXE), el siguiente paso es distribuirlo a sus usuarios.

XLS Padlock ofrece varios métodos para agilizar este proceso: crear un archivo Zip, generar un instalador EXE o copiar directamente los archivos a una carpeta.



Archivo Zip

Un archivo Zip es un método popular y sencillo para distribuir software. Comprimir su aplicación en un único archivo `.zip` simplifica la gestión de archivos para sus usuarios. Este método resulta especialmente útil para compartir su aplicación por correo electrónico o servicios en la nube, ya que reduce el tamaño del archivo y agrupa todos los componentes necesarios en un solo paquete.

Para crear un archivo Zip de su aplicación, haga clic en **Make Zip** y elija dónde desea guardar el archivo `.zip`.

Instalador EXE

Para un toque más profesional, considere distribuir su aplicación con un instalador EXE. Este método mejora la experiencia del usuario al proporcionar un proceso de instalación familiar habitual en los

entornos Windows. Un instalador puede crear accesos directos en el escritorio y en el menú Inicio, mostrar un archivo léame o un acuerdo de licencia, y proporcionar un desinstalador, lo que permite a los usuarios eliminar fácilmente la aplicación de su equipo.

➡ Para obtener instrucciones detalladas, consulte nuestra guía sobre [cómo crear un instalador para su aplicación](#).

Copiar a una carpeta

Por último, copiar los archivos de su aplicación directamente en una carpeta es el método de distribución más directo. Consta de dos sencillos pasos:

- Copie todos los archivos necesarios de la aplicación en una carpeta.
- Comparta esta carpeta con sus usuarios a través de medios físicos (como una unidad USB) o mediante transferencia por red.

Aunque no resulta tan pulido como los otros métodos, este enfoque es la forma más rápida de poner en marcha su aplicación en otro sistema, ya que no requiere utilidades zip especiales ni software de instalación.

👉 Consulte también:

- [Cómo firmar digitalmente su aplicación](#)

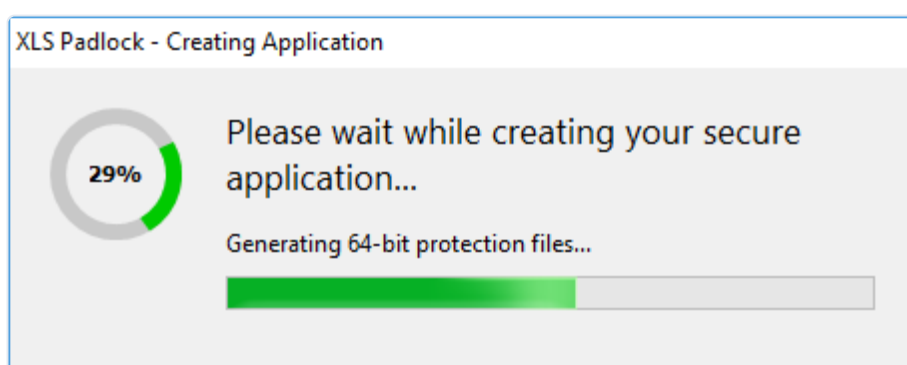
Distribuir un libro protegido

Compile su libro de Excel a EXE

En XLS Padlock, cuando hace clic en “Build Secure Application”



, se muestra el cuadro de diálogo de progreso de la compilación:



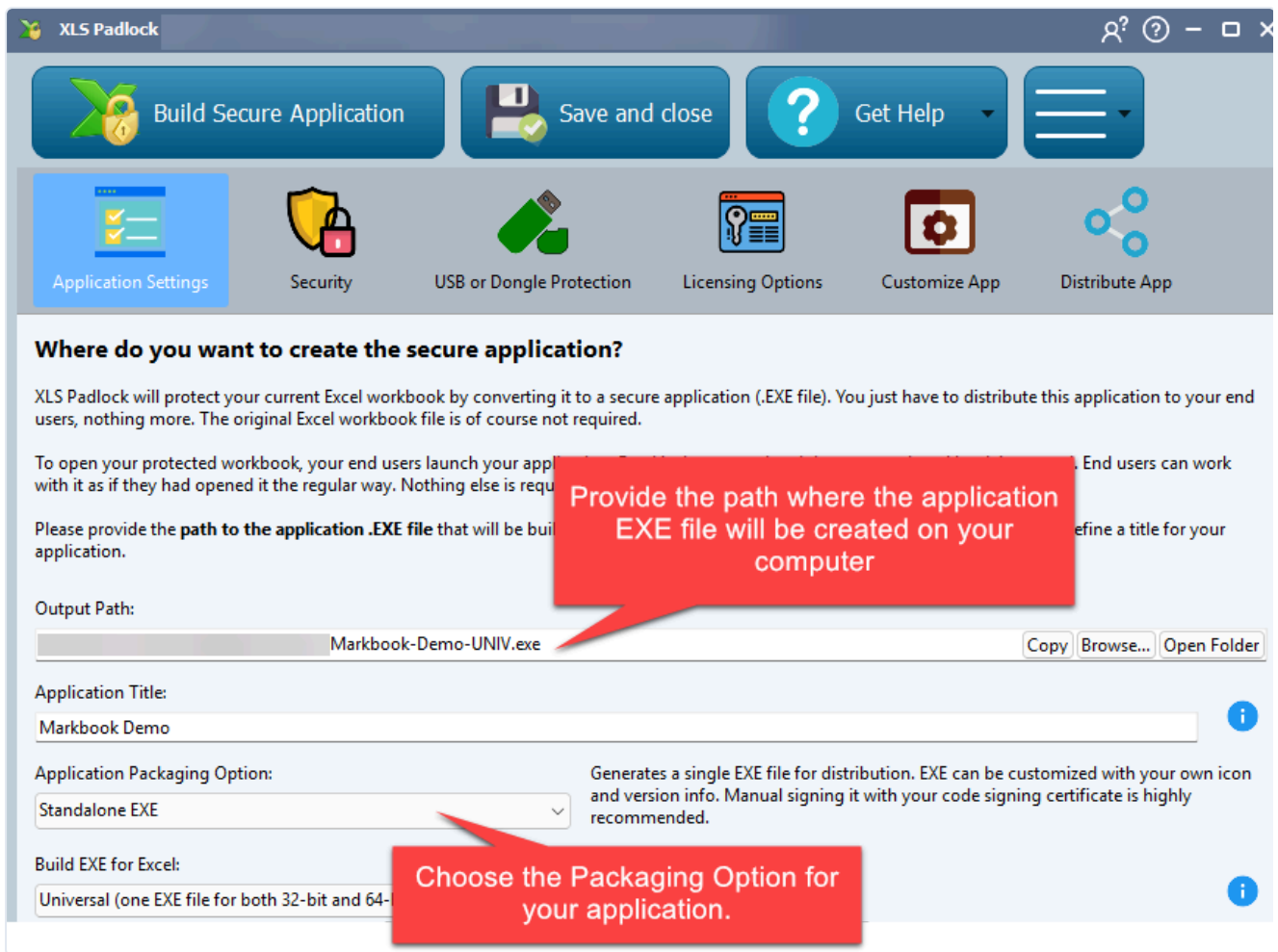
Cuando termina, **su aplicación segura está lista**. XLS Padlock ha generado un **archivo ejecutable** (.EXE) a partir de su libro de Excel.

▶ Ahora, para acceder al libro protegido, los usuarios finales ejecutan este archivo .EXE. Una copia local de Microsoft Excel es el único requisito adicional. Su archivo de libro original **YA NO** es necesario.

- Al ejecutar el archivo .EXE, Excel se inicia y el libro protegido se abre. Los usuarios finales pueden interactuar con él como si lo hubieran abierto de la manera habitual.
- Determinadas funciones de Excel como “New Workbook”, Open y Save (opcionalmente) se desactivan por motivos de seguridad.
- Si no se encuentra Excel en el ordenador del usuario, se mostrará un mensaje de error. Puede configurar su aplicación para [requerir versiones específicas de Excel](#).

Opciones de empaquetado de la aplicación

XLS Padlock ofrece dos formatos para su aplicación segura, lo que influye en cómo la distribuye y gestiona las advertencias de seguridad como Windows SmartScreen.



EXE independiente

XLS Padlock crea un único archivo ejecutable (.EXE) a partir de su libro. Simplemente distribuye este único archivo a sus usuarios finales. El libro de Excel original no es necesario.

La firma de código es muy recomendable

Le aconsejamos encarecidamente [firmar digitalmente sus archivos EXE](#). Distribuir un archivo EXE sin firmar probablemente hará que Windows SmartScreen muestre una advertencia de "Unknown Application". Aunque su EXE seguirá ejecutándose, una firma digital aporta confianza y una mejor experiencia de usuario.

Paquete de aplicación EXE + XPLAPP

Esta opción genera un archivo EXE, un archivo complementario `.bin64` y un archivo de datos `.xplapp` independiente. El EXE principal viene prefirado por nuestra empresa, lo que ayuda al reconocimiento por parte de Windows SmartScreen y del software antivirus.

- Debe distribuir los tres archivos juntos (`.exe` , `.bin64` y `.xplapp`).
- XLS Padlock puede agrupar estos archivos en un [único archivo Zip](#) o un [instalador](#) para facilitar la distribución.

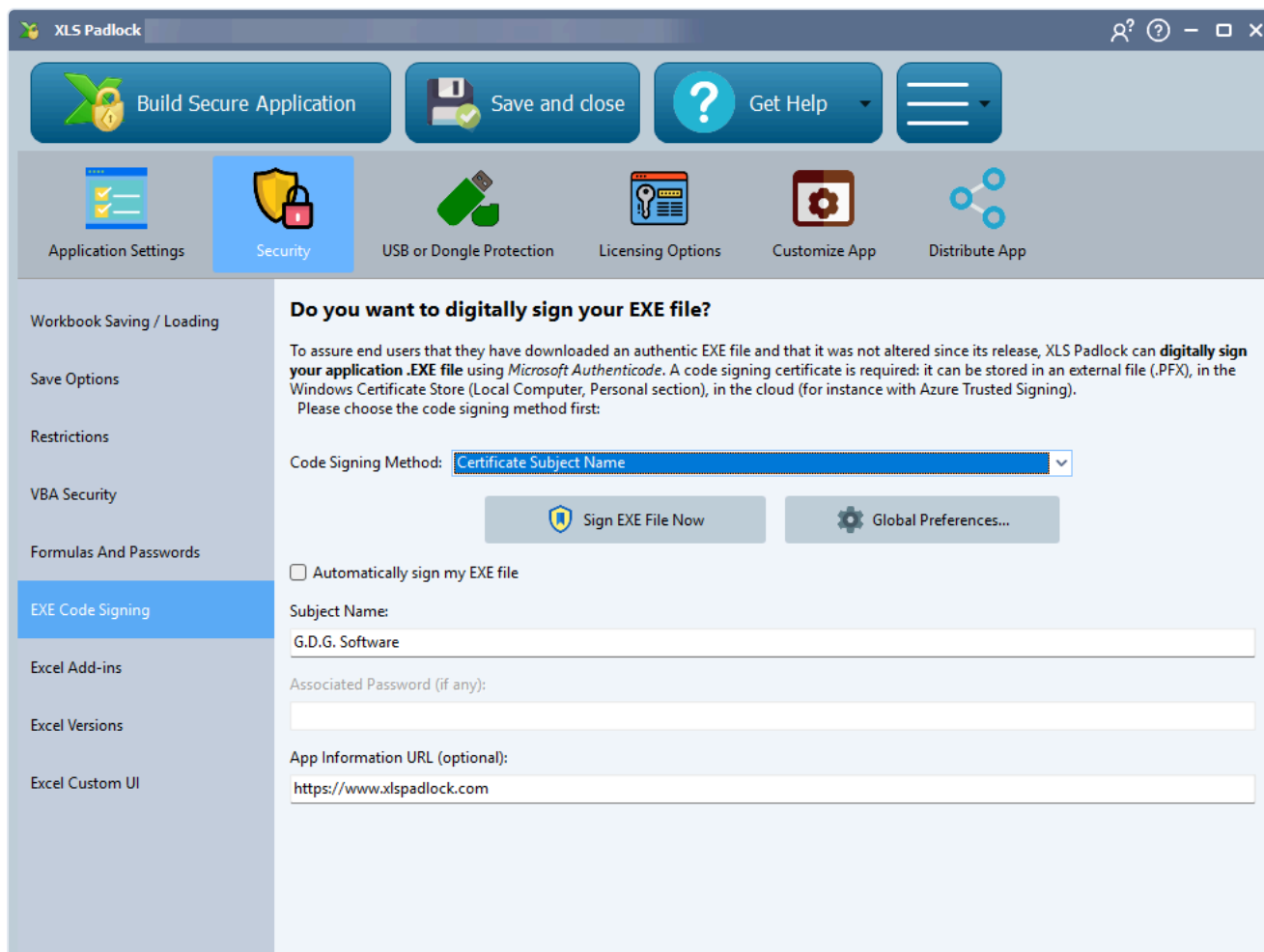
- Esta opción reduce considerablemente el riesgo de advertencias de "Unknown Application" y de falsos positivos de los antivirus.

Recomendaciones de distribución

1. **Firme digitalmente sus archivos EXE:** Este es el paso más importante para evitar falsos positivos de los antivirus y generar confianza con sus usuarios. XLS Padlock puede [automatizar este proceso](#) si dispone de un certificado de firma de código.
2. **Desactive temporalmente su antivirus:** Si tiene un programa antivirus activo, considere desactivarlo antes de compilar sus libros. El software antivirus a veces puede marcar incorrectamente la creación de nuevos archivos EXE en su ordenador.
3. **No envíe archivos EXE por correo electrónico:** La mayoría de los proveedores de correo electrónico bloquean los archivos `.exe`. En su lugar, suba los archivos de su aplicación a un servicio de alojamiento de archivos como Dropbox, Google Drive o su propio servidor web, y comparta el enlace de descarga. También puede [empaquetar su aplicación en un archivo Zip](#).

Firmar digitalmente el EXE

Cuando **firma digitalmente su archivo EXE de libro independiente**, garantiza a los usuarios finales que el archivo es auténtico y no ha sido manipulado. Este proceso, también conocido como firma de código, utiliza la tecnología Microsoft Authenticode® para verificar que el código procede de un editor de confianza.



XLS Padlock simplifica el proceso de firma de código gestionando internamente los pasos necesarios.

Por qué es importante la firma de código

Si tiene previsto distribuir sus aplicaciones a través de Internet, se recomienda encarecidamente la firma de código. Ayuda a evitar que los navegadores web y Windows muestren advertencias de "Unidentified Publisher" y también puede reducir la probabilidad de falsos positivos del software antivirus.

Cómo obtener un certificado de firma de código

Para firmar su aplicación, necesita un **Code Signing Certificate** válido de una autoridad de certificación (CA) de confianza como Sectigo o Digicert. Otros tipos de certificado, como SSL/TLS, no son compatibles.

Certificados basados en la nube

Una alternativa cada vez más popular y rentable es **Azure Trusted Signing**, un servicio basado en la nube de Microsoft. Elimina la necesidad de tokens USB físicos y agiliza el proceso de gestión.

XLS Padlock admite por completo la [firma con Azure Trusted Signing](#).

Certificados basados en tokens (HSM)

Desde el 1 de junio de 2023, todas las claves privadas de los nuevos certificados de firma de código deben almacenarse en hardware seguro, como un token USB compatible con FIPS 140-2 Level 2 o un módulo de seguridad de hardware (HSM). Esto refuerza la seguridad al evitar el robo de claves. XLS Padlock funciona sin problemas con los certificados basados en tokens; solo asegúrese de que el token esté conectado a su ordenador cuando compile su aplicación.

Configurar la firma de código en XLS Padlock

En la interfaz de XLS Padlock, vaya a la pestaña **Security -> EXE Code Signing**. Para habilitar la firma, elija su **Code Signing Method** preferido:

- **PFX File:** Utiliza un certificado almacenado en un archivo `.pfx`. Este es un método heredado para certificados antiguos.
- **Certificate Subject Name:** Localiza el certificado en el almacén de certificados de Windows por su nombre de sujeto (Subject Name). Este es un método habitual para certificados en tokens de hardware.
- **Certificate Thumbprint:** Localiza el certificado en el almacén de certificados de Windows por su huella digital única (Thumbprint, un hash SHA-1). A menudo es el método más fiable.
- **SignTool Commands:** Un método avanzado que le permite proporcionar comandos personalizados para la utilidad `SignTool.exe` de Microsoft, ofreciendo la máxima flexibilidad.
- **Azure Trusted Signing:** Firma su aplicación con el servicio basado en la nube de Microsoft. Consulte nuestro [tutorial de Azure Trusted Signing](#) para obtener más detalles.

Azure CLI Required

Para usar este método, primero debe instalar la CLI de Microsoft Azure e iniciar sesión con `az login`.

Realizar la firma

- **Firma manual:** Haga clic en **Sign EXE File Now** para firmar de inmediato el último archivo EXE compilado.

- **Firma automática:** Marque **Automatically sign my EXE file** para que XLS Padlock firme el EXE cada vez que compile su aplicación.

Resolución de problemas

Si se produce un error de firma, consulte los mensajes detallados en el registro de compilación de XLS Padlock. Este archivo de registro suele denominarse [Your Workbook Filename].xplcompil.log y se encuentra en el mismo directorio que su libro.

Crear un instalador

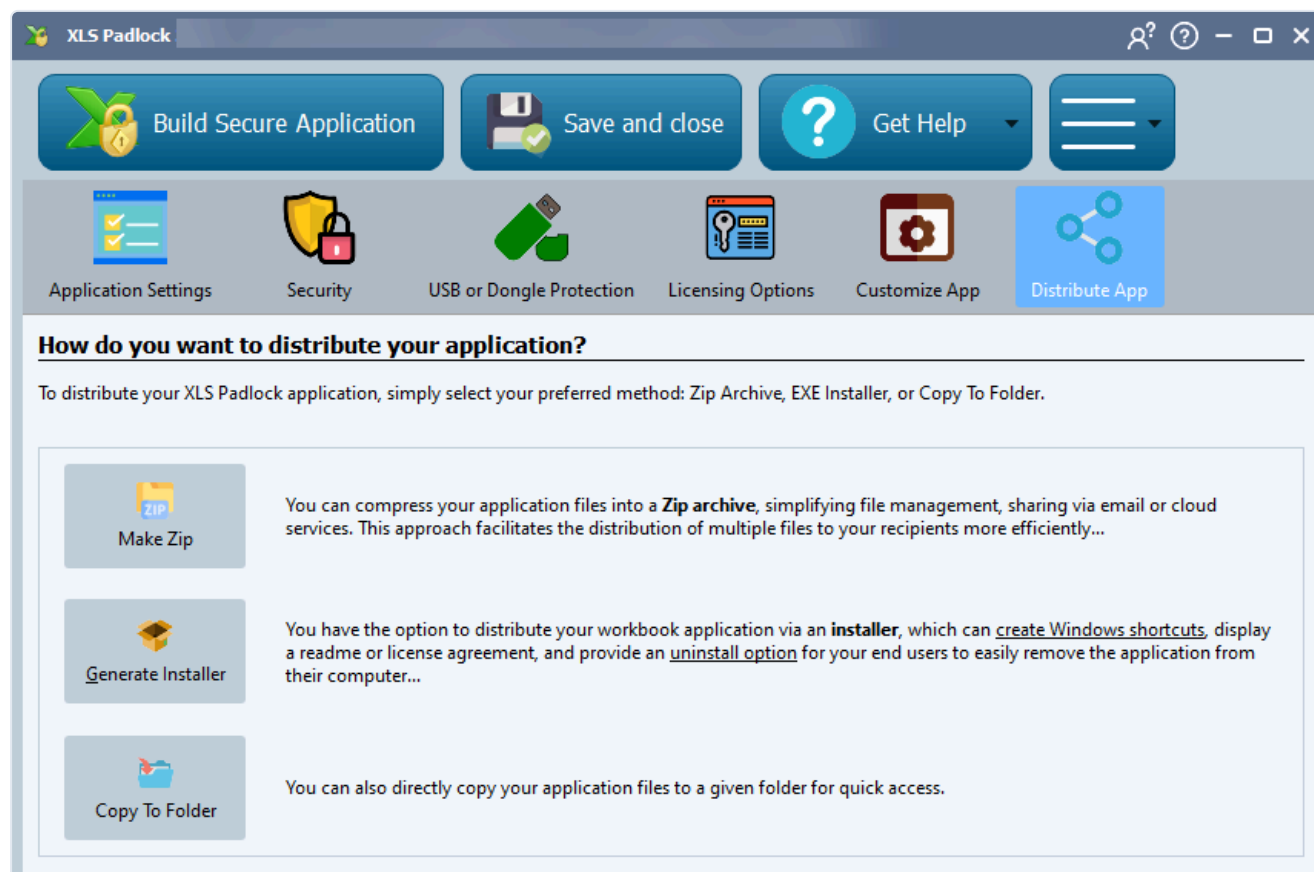
Empaquetar su libro compilado en un instalador profesional (o programa de instalación) es una forma excelente de distribuir su aplicación. Los instaladores ofrecen una experiencia de usuario familiar y pueden encargarse de tareas como instalar varios archivos (como un archivo léame), mostrar un acuerdo de licencia, crear accesos directos y proporcionar un desinstalador para una eliminación limpia.

XLS Padlock se integra con nuestro software, [Paquet Builder](#), para generar instaladores personalizados y compactos. **Paquet Builder debe estar instalado en su ordenador para utilizar esta función.**

Paquet Builder es una potente herramienta que combina un creador de archivos autoextraíbles 7-Zip con un generador de rutinas de instalación. Le permite crear instaladores flexibles y compactos para la distribución de software profesional, empaquetar documentos y archivos de programa, crear paquetes de instalación multilingües, generar actualizaciones y agrupar varios archivos en un único `.exe` para facilitar su distribución en línea.

Para crear un instalador:

1. Haga clic en **Generate Installer** en la pestaña "Distribute" de XLS Padlock. Esto abrirá la ventana "Make Setup for your compiled workbook".



1. Rellene "Destination path" (carpeta de instalación predeterminada), "Setup title" (título de la ventana del instalador) y "Your Application Name" (para accesos directos, desinstalador, etc.).
2. Pulse **Generate Installer** para crear el proyecto de Paquet Builder. A continuación, XLS Padlock iniciará Paquet Builder, lo que le permitirá personalizar aún más el proyecto y compilar el instalador

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):

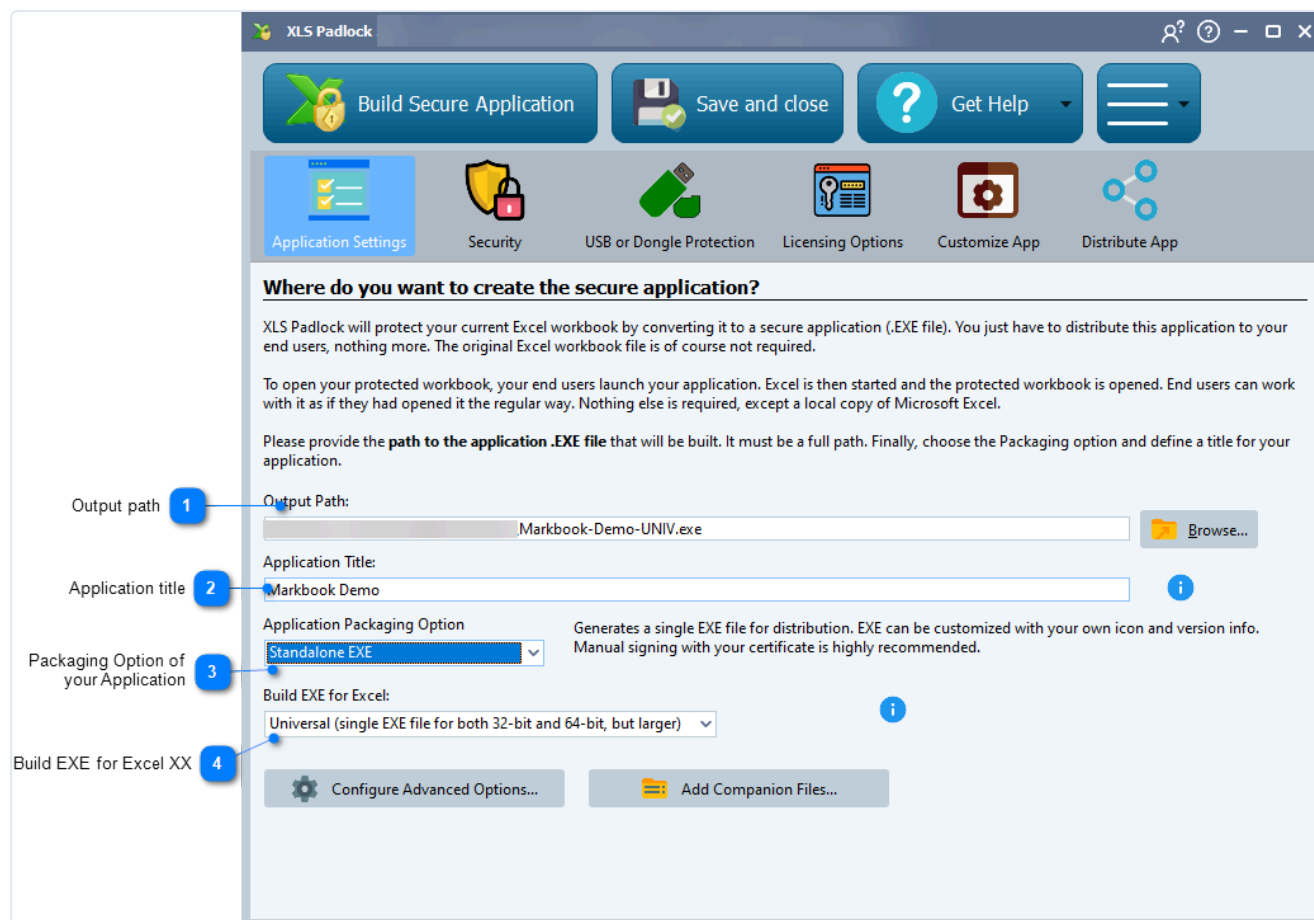
Open the generated project in Paquet Builder

 **Generate Installer**
 **Close**
 **Help**

👉 Consulte el [tutorial completo sobre cómo crear un instalador para su libro de Excel](#).

Compilar el EXE según la arquitectura de Excel

Microsoft Excel está disponible en dos arquitecturas: 32 bits y 64 bits. Como 64 bits es ahora el estándar para las versiones modernas de Office, la mayoría de sus usuarios necesitará una aplicación compatible con 64 bits. Sin embargo, algunos pueden seguir usando la versión de 32 bits, especialmente en sistemas más antiguos. XLS Padlock le permite compilar para ambas plataformas.



Esta opción solo está disponible para los [archivos ejecutables independientes](#).

👉 Elija el tipo de archivo EXE que generará XLS Padlock:

- **32-bit only**: Genera un archivo EXE compatible solo con las versiones de 32 bits de Excel.
- **64-bit only**: Genera un archivo EXE compatible solo con las versiones de 64 bits de Excel.
- **Universal**: Crea un único archivo EXE que funciona con las versiones de 32 bits y de 64 bits de Excel. Esto simplifica la distribución, pero el archivo EXE resultante tendrá aproximadamente el doble de tamaño.
- **32-bit and 64-bit**: Genera dos archivos EXE independientes de forma simultánea. Para evitar que se sobrescriban entre sí, XLS Padlock añade automáticamente los sufijos "32" y "64" a los nombres de archivo (por ejemplo, `MyApp32.exe` y `MyApp64.exe`).

Discrepancia de versión

Si un usuario final ejecuta un EXE con una arquitectura de Office no coincidente (por ejemplo, un EXE de 64 bits en una instalación de Office de 32 bits), recibirá un mensaje de error indicándole que se ponga en contacto con usted para obtener la versión correcta.

Se recomienda la firma de código para el modo Universal

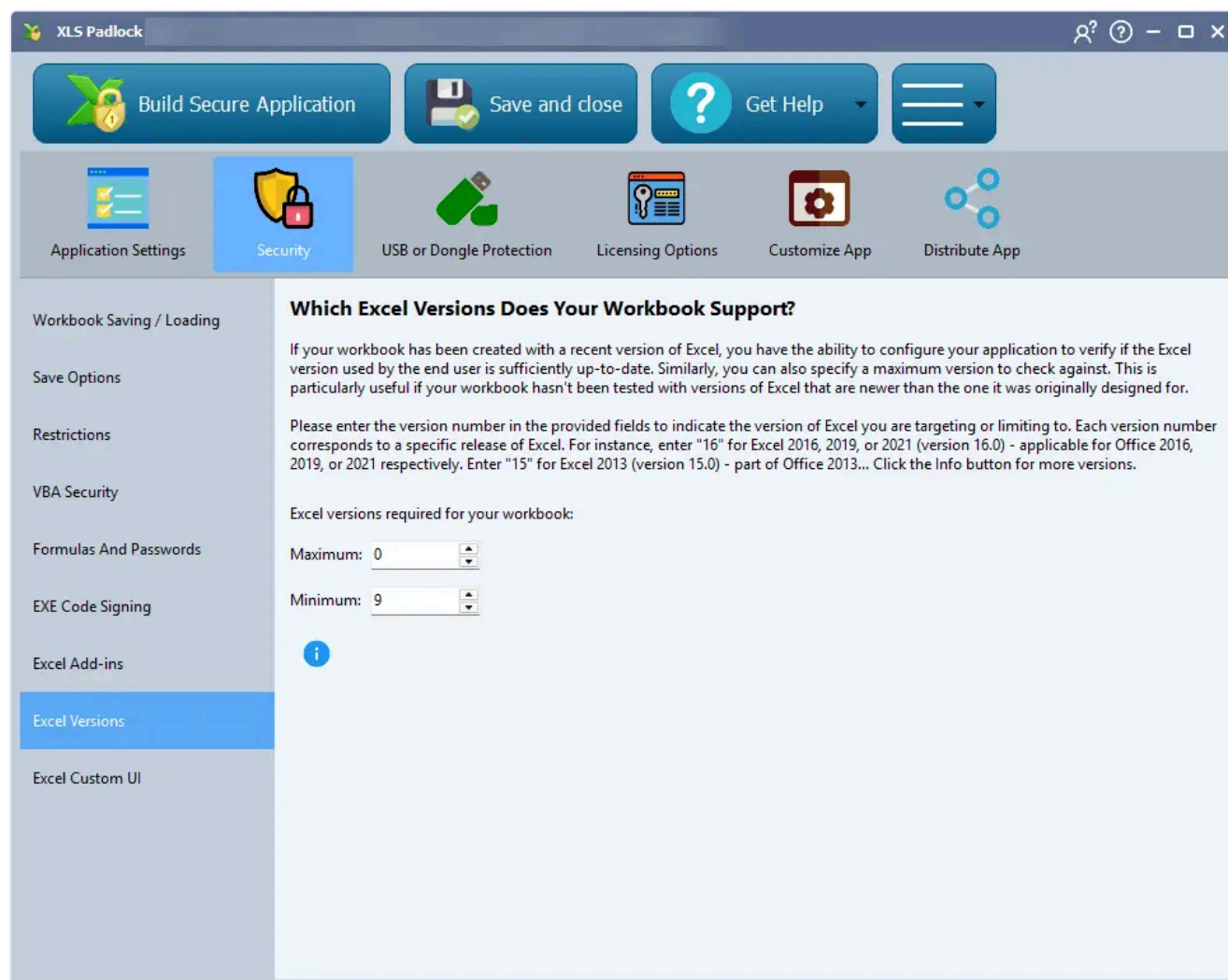
Si utiliza el modo Universal, le recomendamos encarecidamente que firme digitalmente su aplicación. Esto se debe a que se crea un archivo EXE temporal en tiempo de ejecución, que a veces puede ser detectado por el software antivirus. Una firma digital reduce considerablemente el riesgo de falsos positivos.

Información

De forma predeterminada, las [aplicaciones distribuidas en formato de paquete \(bundle\)](#) ya son universales.

Versiones de Excel

Si su libro requiere funciones de una versión específica de Excel, puede configurar su aplicación para comprobar la versión de Excel del usuario final en tiempo de ejecución.



Puede establecer tanto una versión mínima requerida como una versión máxima permitida. Esto resulta útil para garantizar la compatibilidad o para impedir el uso en versiones de Excel más recientes y no probadas.

Introduzca el número de versión correspondiente en los campos:

- **9:** Excel 2000 (versión 9.0)
- **10:** Excel 2002 (versión 10.0)
- **11:** Excel 2003 (versión 11.0)
- **12:** Excel 2007 (versión 12.0)
- **14:** Excel 2010 (versión 14.0)
- **15:** Excel 2013 (versión 15.0)
- **16:** Excel 2016, 2019, 2021 y 365 (versión 16.0)

Actualizaciones del libro

Después de distribuir su aplicación de libro de Excel, tarde o temprano necesitará proporcionar actualizaciones a sus usuarios. XLS Padlock ofrece varias herramientas para agilizar este proceso.

Una consideración importante para las actualizaciones es cómo guarda los datos del usuario su aplicación. Debe conocer los [distintos modos de guardado que ofrece XLS Padlock](#) antes de compilar su aplicación.

Si su libro se va a actualizar con frecuencia y sus usuarios solo necesitan modificar unos pocos valores de celda, el modo **Save defined cell values only (.XLSCE file)** es la mejor opción. Este modo le permite actualizar el archivo principal del libro sin afectar a los datos del usuario, ya que XLS Padlock guarda y restaura [únicamente los valores de celda que usted haya designado](#).

Cuando actualice su libro de origen, la forma más directa de distribuir la actualización es recompilarlo y enviar el nuevo archivo EXE a sus clientes.

Flujo de trabajo de actualización recomendado

El comportamiento en el extremo del usuario depende de lo que haya cambiado entre versiones:

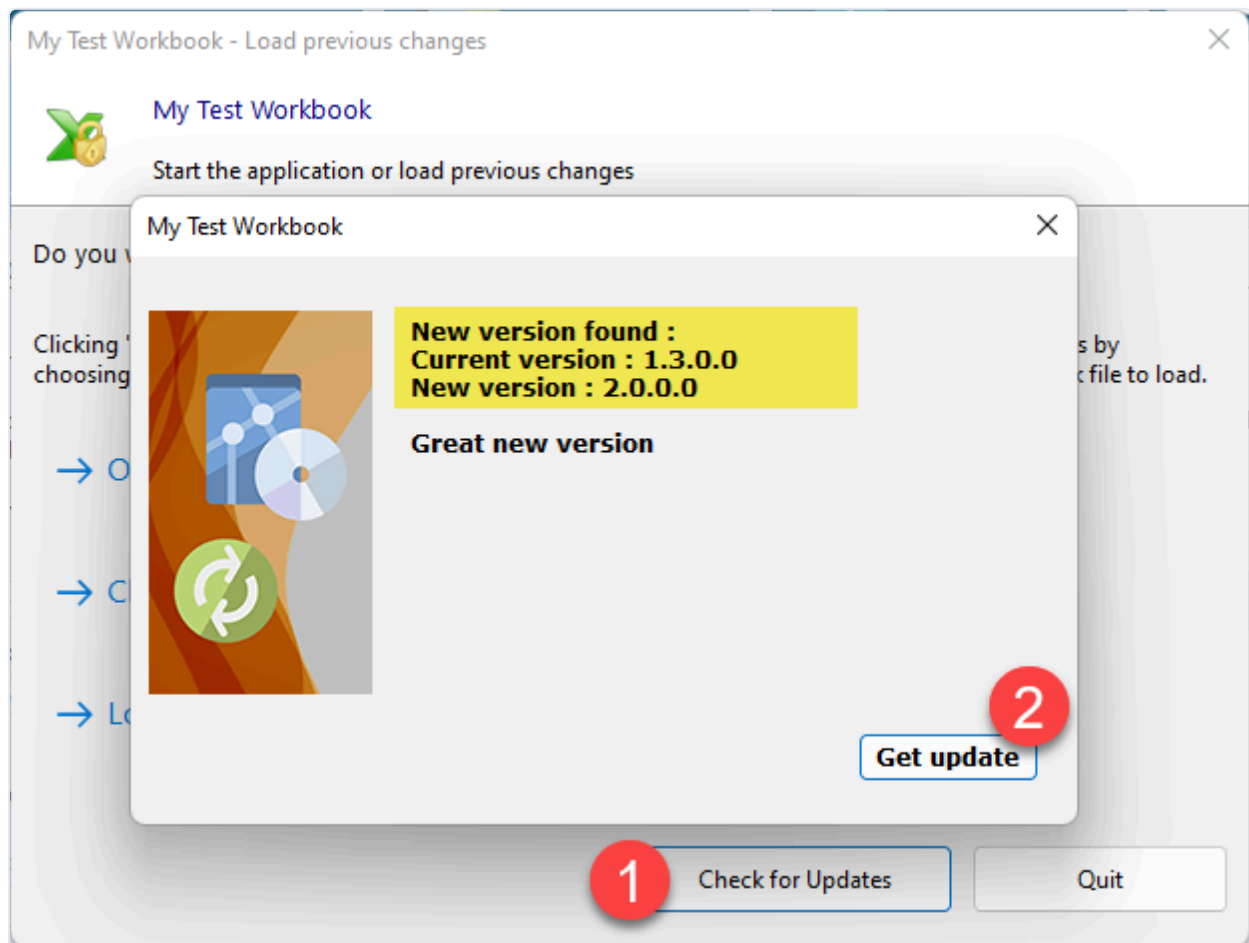
- **Solo el contenido del libro** (fórmulas, formato, hojas, lógica VBA): conserve el **Application GUID** y la **Secret Key** existentes. En el modo `.XLSCE`, los usuarios verán su nueva lógica con sus valores de celda guardados restaurados. En el modo `.XLSC`, los usuarios seguirán viendo su última instantánea guardada; sus cambios solo aparecerán si eligen manualmente "Original Workbook" en la pantalla de bienvenida.
- **Desea invalidar por completo los antiguos archivos de guardado del usuario**: genere una nueva **Secret Key** en la configuración de [Activation and Licensing](#), pero deje el **Application GUID** sin cambios para que las claves de activación existentes sigan siendo válidas. En el primer arranque del usuario, su antiguo archivo de guardado se detectará como incompatible y se le pedirá, a través de la pantalla de bienvenida, que cargue en su lugar el libro original incrustado.
- **No cambie de modo de guardado entre versiones** (`.XLSC` ↔ `.XLSCE`). Los archivos de guardado existentes del modo anterior se ignorarán, y se perderán todos los datos que sus usuarios aún no hubieran vuelto a introducir. Elija un modo de guardado al principio del proyecto y manténgalo durante toda la vida útil de la aplicación.

CONSEJO

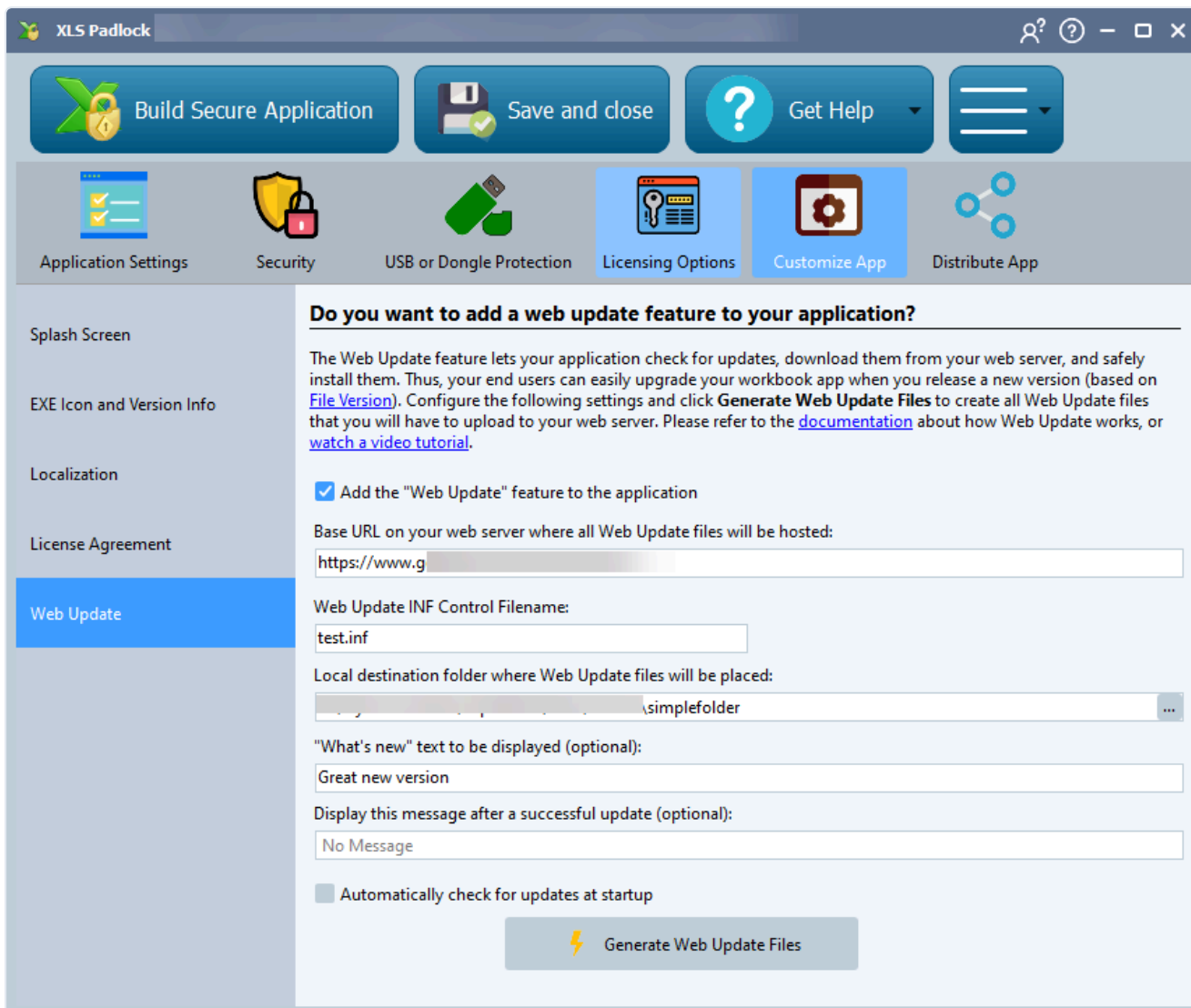
El [botón Original Workbook de la pantalla de bienvenida](#) actúa como red de seguridad del usuario cuando un antiguo archivo de guardado no se puede cargar. Aunque lo haya ocultado, XLS Padlock lo volverá a habilitar automáticamente para el intento de reintento, de modo que los usuarios nunca queden bloqueados.

👉 Sin embargo, descargar y reemplazar manualmente el archivo EXE puede resultar tedioso para los usuarios finales. Para simplificarlo, XLS Padlock ofrece una función de actualización web automática.

Puede configurar su aplicación para que compruebe si hay actualizaciones en línea y notifique a los usuarios cuando haya una nueva versión disponible:



Esta función se puede configurar rápidamente y directamente dentro de XLS Padlock:



👉 Para configurar su propio sistema de actualización, necesitará un servidor web o un espacio de alojamiento donde pueda colocar archivos para su descarga directa. A continuación, siga las instrucciones de la página [cómo configurar las actualizaciones web automáticas](#).

Tutorial en vídeo

Vea nuestro tutorial en vídeo sobre [cómo configurar actualizaciones web automatizadas para sus libros de Excel](#).

Actualizaciones web automáticas

XLS Padlock incluye una función de actualización web que puede descargar e instalar automáticamente nuevas versiones de su aplicación. Esto permite que sus usuarios finales actualicen fácilmente la aplicación de su libro cada vez que usted publique una nueva versión.

No se requiere ningún software de terceros, pero **debe disponer de un servidor web o de un espacio de alojamiento web** para alojar los archivos de actualización.

Tutorial en vídeo

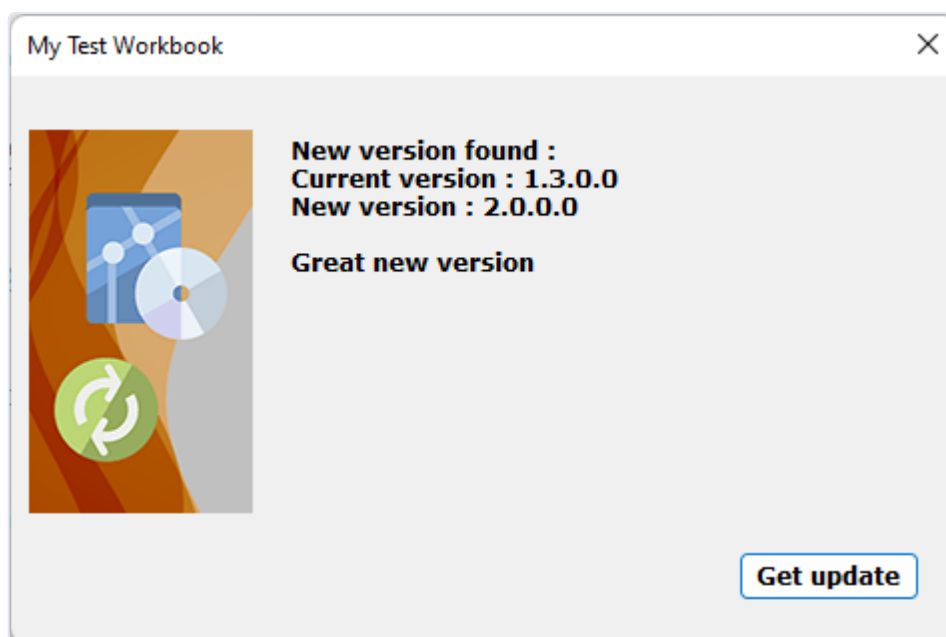
Vea un tutorial en vídeo sobre cómo configurar las actualizaciones web automáticas para libros de Excel.

Cómo funciona la actualización web

La aplicación descarga primero un pequeño archivo de control (`.inf`) desde su servidor. Este archivo indica a la aplicación si hay disponible una nueva versión. Si la hay, la aplicación descarga los archivos de actualización necesarios (`.cab`), se cierra, instala la actualización y se reinicia.

Comprobación de actualizaciones

Puede configurar su aplicación para que **compruebe automáticamente si hay actualizaciones al iniciarse** o añadir un botón "**Check for Updates**" a la [pantalla de bienvenida](#). Cuando se detecta una nueva versión, se guía a los usuarios a través de un asistente de actualización web.



Los usuarios también pueden iniciar una actualización mediante el [modificador de línea de comandos](#) `webupdate` .

Cómo se detecta una nueva versión

El número de **File Version**, especificado en la página [EXE Version Info](#) de XLS Padlock, se utiliza para determinar si es necesaria una actualización. Debe aumentar este número de versión cada vez que publique una actualización.

Configuración

- **Base URL:** La URL de su servidor web donde se alojarán los archivos de actualización web (por ejemplo, `https://www.yourwebsite.com/myfolder`). Se recomienda HTTPS.
- **Web Update INF Control Filename:** El nombre del archivo de control que contiene la información de actualización.
- **Local Destination Folder:** Una carpeta local en su equipo donde XLS Padlock generará los archivos de actualización. A continuación, debe subir todo el contenido de esta carpeta a su servidor web.

Advertencia

La carpeta de destino debe estar vacía. XLS Padlock pedirá eliminar su contenido si no lo está.

- **"What's new" text (opcional):** Texto que se mostrará en el asistente de actualización, informando a los usuarios sobre los cambios de la nueva versión.

Generación de los archivos de actualización web

Al hacer clic en **Generate Web Update Files** en XLS Padlock se crearán los archivos necesarios (un archivo de control `.inf` y un archivo comprimido `.cab`) en la carpeta local que haya especificado. A continuación, puede subir estos archivos a su servidor web.

Privilegios de administrador

Si la aplicación está instalada en una carpeta restringida de Windows como `Program Files` , se requerirán privilegios de administrador para instalar la actualización, lo que activará un aviso de UAC.

Resolución de problemas

Para resolver problemas con la función de actualización web, active la opción **"Enable WebUpdate Log"** en [Opciones avanzadas](#). Esto creará un archivo `WUPDATE.LOG` en el directorio Documentos del usuario con información detallada sobre el proceso de actualización.

Migrar los datos de usuario entre actualizaciones

Cuando actualiza su libro protegido y distribuye un nuevo EXE, podría afrontar dificultades para garantizar que los datos de sus usuarios se transfieran correctamente, especialmente si utiliza el [modo Full Save](#). Este modo crea una **instantánea completa del libro** en el momento de guardar. En consecuencia, cuando un usuario abre su antiguo archivo guardado (`.xlsc`) con la nueva aplicación, verá sus datos antiguos en la antigua estructura del libro y no se beneficiará de sus últimas actualizaciones (como nuevas funciones o correcciones de errores en su código VBA).

Este tema explica cómo utilizar un sistema de exportación/importación basado en VBA para ayudar a sus usuarios a migrar sus datos de una versión antigua de su aplicación a una nueva.

Advertencia

La solución que aquí se describe depende del [modo de guardado que haya seleccionado para su proyecto de XLS Padlock](#). Si utiliza el modo de guardado **Cell Values**, los datos del usuario se cargan automáticamente en la estructura de su libro actualizado, y este proceso manual no es necesario.

Alternativa: modo de guardado Cell Values

Como se ha mencionado anteriormente, el [modo de guardado Cell Values](#) es la forma más sencilla de gestionar las actualizaciones. Sin embargo, este modo está diseñado para guardar solo los valores de celdas específicas. Si necesita transferir otros tipos de datos, como el estado de variables u objetos de VBA, tendrá que utilizar una solución personalizada de VBA como la que se describe en este tema.

Ejemplo de código

Este proceso consta de dos pasos principales: exportar los datos de la versión antigua e importarlos a la nueva.

1. Exportar los datos del usuario a un archivo de Excel

La macro siguiente generará un archivo de Excel `.xlsx` normal que contiene los datos del usuario. Debe incluir esta macro en su antigua aplicación, vinculada a un botón, para que los usuarios puedan exportar sus datos antes de actualizar.

NOTA

Tendrá que sustituir los nombres de las hojas y los rangos de celdas en la macro siguiente para que coincidan con la estructura de su libro.

```
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. Cargar los datos en el nuevo archivo EXE.

El usuario debe abrir el nuevo EXE y ejecutar la 3.^a macro siguiente para cargar los datos (vincule la macro a un botón). Una vez que el usuario ejecute la macro, se le pedirá que seleccione el archivo (1.^a macro), y los datos se copiarán (2.^a macro). La tercera macro ejecutará ambas macros, y esa es la macro que debe vincularse al botón. De nuevo, tendrá que cambiar los nombres de las celdas y de las pestañas en la segunda macro, y también puede cambiar el título en la primera macro:

```
***** 1st macro:
Sub Open_Workbook_Dialog()
Dim my_FileName As Variant
my_FileName = Application.GetOpenFilename( _
FileFilter:="Excel Files,.xl;.xm", _
FilterIndex:=3, _
Title:="Select the old version of your file, where you will pull the data from", _
MultiSelect:=False)
If my_FileName <> False Then
Workbooks.Open Filename:=my_FileName
End If
End Sub

***** 2nd macro:
Sub TransferData()
If Workbooks.Count > 1 Then
Workbooks(1).Sheets("SheetA").Range("A1:C3").Value = Workbooks(2).Sheets("SheetA").Range("A1:C3").Value
Workbooks(1).Sheets("SheetB").Range("B3").Value = Workbooks(2).Sheets("SheetB").Range("B3").Value
Workbooks(1).Sheets("SheetC").Range("B1:C3").Value = Workbooks(2).Sheets("SheetC").Range("B1:C3").Value
Workbooks(2).Close savechanges:=False
Else
MsgBox "The data hasn't been transferred.", vbExclamation, "Error"
End If
End Sub

***** 3rd macro:
Sub TheTransfer()
Call Open_Workbook_Dialog
Call TransferData
End Sub
```

NOTA

Para guardar libros normales, también puede consultar: [Cargar/guardar libros mediante el asistente VBA SetOption](#)

👉 **Vea también:**

- [Modo de guardado que ha seleccionado para su proyecto de XLS Padlock](#)

Guardar y restaurar la configuración con plantillas

XLS Padlock le permite guardar la configuración actual de su proyecto en un archivo de plantilla. Es una función muy útil que puede ahorrarle una cantidad considerable de tiempo, especialmente si crea con frecuencia nuevas aplicaciones de libros con configuraciones similares.

Guardar la configuración del proyecto en una plantilla

Una vez que haya configurado todas las opciones deseadas para su proyecto (como la configuración de seguridad, la personalización y las opciones de salida), puede guardar esta configuración como una plantilla.

Haga clic en el botón



App Menu y, a continuación, en el menú **Save Current Settings As Template**. Se le pedirá que elija una ubicación y un nombre para su archivo de plantilla (que tendrá la extensión `.XPLP`).

Advertencia

No elija su archivo de proyecto de XLS Padlock existente.

Restaurar la configuración desde una plantilla

Cuando inicie un nuevo proyecto o desee aplicar una configuración estándar a uno existente, puede restaurar fácilmente la configuración desde una plantilla guardada previamente.

Haga clic en el botón



App Menu, luego en el menú **Restore Project from Template**, navegue hasta su archivo de plantilla `.XPLP` y selecciónelo. Toda la configuración almacenada en esa plantilla se aplicará instantáneamente a su proyecto actual.

NOTA

La [ruta de salida de la aplicación](#) no se almacena dentro de las plantillas.

Error de registro o EREGISTRYEXCEPTION

Este error suele indicar un conflicto con un software de seguridad o un programa antivirus. Para resolverlo, active la opción "**Use another registry key for storing activation data**" en las [Opciones avanzadas](#).

Violación de acceso en la dirección

Este error puede producirse al usar la función "Lock VBA Project (simple VBA protection)". Para resolverlo, quite la contraseña de su proyecto VBA antes de compilar el libro con XLS Padlock. La contraseña no es necesaria, ya que XLS Padlock bloqueará el proyecto automáticamente sin solicitar ninguna.

Como alternativa, puede probar a usar una contraseña más corta o cambiar a la opción [Prevent access to VBA editor](#).

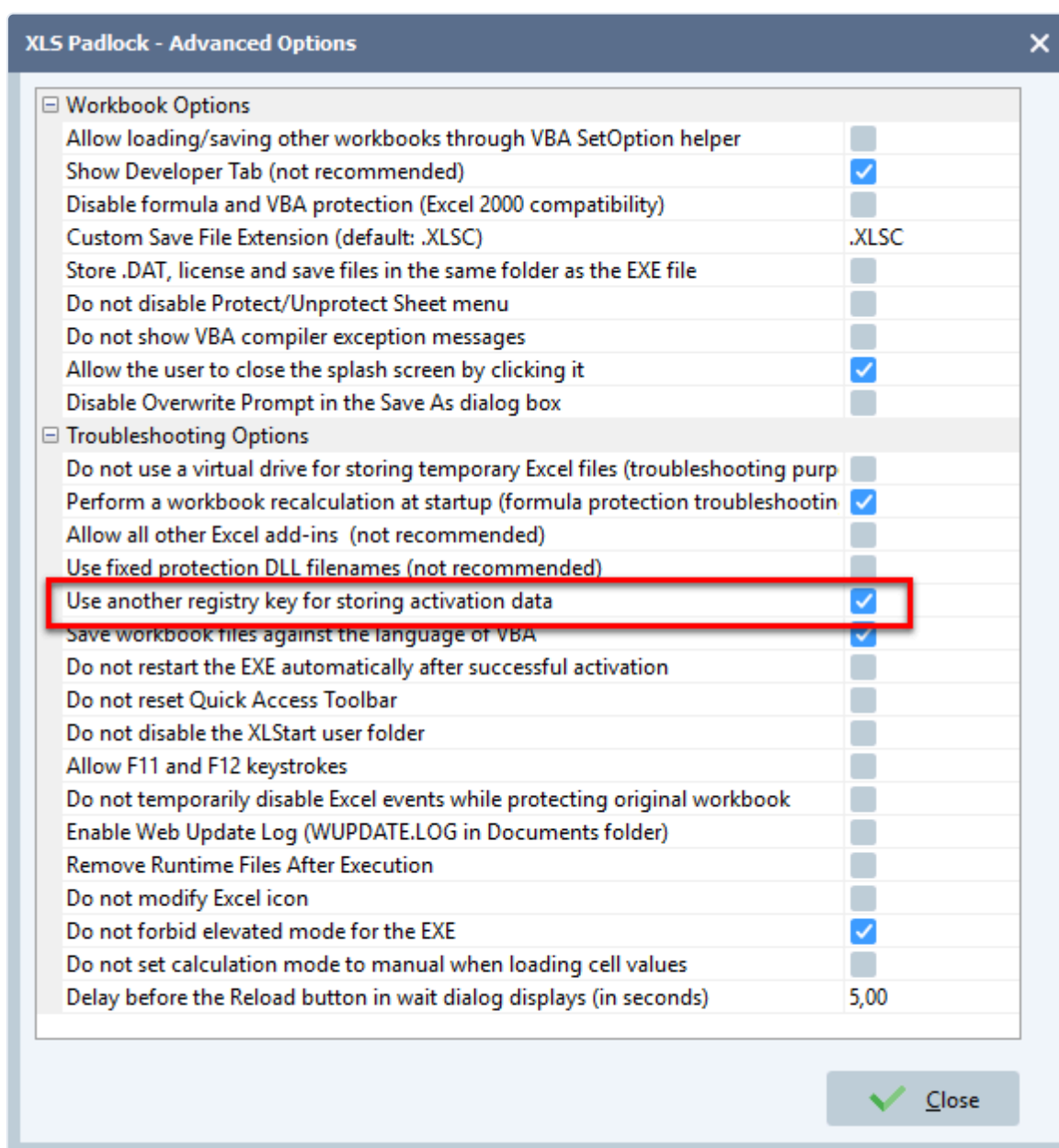
Error «Failed to set data for 'Data'»

Si su usuario final recibe el siguiente mensaje de error al introducir una clave de activación, normalmente indica un conflicto con un software de seguridad en su sistema.

Error setting data in registry, Failed to set data for 'Data'

Este error se produce porque otro programa, normalmente un antivirus o una suite de seguridad, impide que la aplicación escriba sus datos de activación en el Registro de Windows.

Para resolverlo, puede configurar su proyecto de XLS Padlock para que use una clave del registro alternativa. Active la opción avanzada **Use another registry key for storing activation data**, como se muestra a continuación, y vuelva a compilar su aplicación.



Por qué el EXE es tan grande

El archivo `.exe` compilado es considerablemente más grande que el libro original porque es una aplicación autónoma que agrupa varios componentes esenciales:

- Un cargador seguro compatible con varias versiones de Excel y Windows.
- Un motor de ejecución completo compatible con Unicode.
- El compilador e intérprete de VBA propietario.
- Múltiples capas de protección contra la piratería y del código.

Para reducir el tamaño del archivo EXE:

- **Use UPX Compression:** UPX es un compresor de archivos EXE gratuito y de código abierto disponible en upx.github.io. Comprimir el EXE puede reducir su tamaño de forma significativa. No obstante, recomendamos encarecidamente firmar digitalmente cualquier archivo EXE comprimido, ya que de lo contrario algunos programas antivirus podrían generar falsos positivos.
- **Disable Formula Protection:** Si su aplicación no depende de fórmulas sensibles, puede desactivar la protección de fórmulas para reducir ligeramente el tamaño final del archivo EXE.

Restaurar el archivo XLS desde el EXE

No, no es posible restaurar el archivo de Excel original a partir de un .EXE compilado si la función "Save As" estaba desactivada.

Si necesita poder recuperar el libro original, debe activar la función "Save As" durante la compilación. Esto le permite usar la funcionalidad de [descifrado de archivos guardados](#).

Enlaces de soporte

Si tiene alguna pregunta o problema con XLS Padlock, puede ponerse en contacto con nosotros a través de los siguientes canales:

- **Foro de usuarios:** <https://www.gdgsoft.info> (Elija la categoría "XLS Padlock")
- **Soporte por correo electrónico:** info@xlspadlock.com
- **Para soporte de Enky SL/LC Dongle:** info@hs-securityware.com

Al solicitar soporte, incluya su versión de Excel (por ejemplo, Excel 365, Excel 2019) y su arquitectura (32 bits o 64 bits) para ayudarnos a resolver su problema más rápido.

Síguenos en X: <https://x.com/gdgsoft>

Nuestros otros productos están disponibles en: <https://www.gdgsoft.com>