

XLS PADLOCK

Guia do usuário do XLS Padlock

Proteção, licenciamento e distribuição de planilhas
Excel

Version 2026.0

G.D.G. Software

www.xlspadlock.com

Copyright © G.D.G. Software 2018-2026. Todos os direitos reservados.

Índice

- 01 Sobre este guia

- 02 Introdução ao XLS Padlock

- 03 Baixar e instalar

- 04 Configurações do aplicativo

- 05 XLS Padlock Manager

- 06 Como proteger uma planilha do Excel

- 07 Segurança

- 08 Opções de nuvem e IA

- 09 Melhorar a proteção

- 10 Controle de acesso à planilha

- 11 Restrições da planilha do Excel

- 12 Proteção por senha da planilha

- 13 Combinar a proteção do Excel e do XLS Padlock

- 14 Proteção de fórmulas: Excel versus XLS Padlock

- 15 Proteger fórmulas com o XLS Padlock

- 16 Marcar células para proteção de fórmulas

- 17 Método de proteção de fórmulas

- [18](#) Desativar a proteção de fórmulas

- [19](#) Sobre o VBA Compiler integrado

- [20](#) Proteção real do código VBA

- [21](#) Referência de sintaxe

- [22](#) Livro de receitas da API VBA

- [23](#) Escrever e compilar VBA seguro

- [24](#) Invocar o VBA compilado em tempo de execução

- [25](#) Acessar objetos do Excel

- [26](#) Passar matrizes

- [27](#) Passar mais parâmetros

- [28](#) Tratar erros

- [29](#) OLE error 800A03EC

- [30](#) Proteção do código VBA

- [31](#) Ocultar e bloquear o código VBA

- [32](#) Bloquear o projeto VBA

- [33](#) Proibir o acesso ao VBE

- [34](#) Guia de ativação e licenciamento

- [35](#) Chaves de ativação

- [36](#) Configurar chaves de ativação

37 Chaves de ativação com bloqueio por hardware

38 Ativação on-line

39 Validação on-line

40 Editor do formulário de registro

41 Mostrar o contrato de licença

42 Gerador de chaves (portátil e servidor)

43 SDK do gerador de chaves

44 Gerador de chaves independente

45 Restrições nas chaves

46 Proteção por USB ou dongle

47 Desativação

48 Criar planilhas de avaliação

49 Fechar após um tempo determinado

50 Verificar o estado de avaliação

51 Dias de avaliação restantes

52 Opções de salvamento da planilha

53 Salvar e carregar

54 Modo de salvamento: completo ou valores de células

55 Definir células a salvar e restaurar

- 56 Restaurar e salvar com VBA

- 57 Acessar a planilha segura e os arquivos complementares

- 58 Adicionar arquivos complementares

- 59 Pasta de armazenamento de salvamentos

- 60 Armazenar as alterações dentro do EXE

- 61 Restringir o carregamento e o salvamento

- 62 Abrir e descriptografar os salvamentos

- 63 Bloquear os salvamentos a uma máquina

- 64 Referências externas e hiperlinks

- 65 Obter o caminho ao lado da planilha compilada

- 66 Personalizar o aplicativo

- 67 Empacotamento do aplicativo

- 68 Tela de abertura

- 69 Alterar o ícone do EXE

- 70 Janela do Excel ao iniciar

- 71 Executar como aplicativo somente VBA

- 72 Informações de versão do EXE

- 73 Opções de linha de comando

- 74 Ocultar a caixa de diálogo de carregamento

- [75](#) Localização e tradução

- [76](#) Opções avançadas

- [77](#) Ignorar erros de processamento

- [78](#) Desativar informações de depuração

- [79](#) Interface personalizada do Excel

- [80](#) Suplementos do Excel

- [81](#) Distribuir o aplicativo

- [82](#) Distribuir uma planilha protegida

- [83](#) Assinar digitalmente o EXE

- [84](#) Criar um instalador

- [85](#) Compilar o EXE conforme a arquitetura do Excel

- [86](#) Versões do Excel

- [87](#) Atualizações da planilha

- [88](#) Atualizações web automáticas

- [89](#) Migrar os dados do usuário entre atualizações

- [90](#) Salvar e restaurar as configurações por meio de modelos

- [91](#) Erro de registro ou EREGISTRYEXCEPTION

- [92](#) Violação de acesso no endereço

- [93](#) Erro "Failed to set data for 'Data'"

94 Por que o EXE é tão grande

95 Restaurar o arquivo XLS a partir do EXE

96 Links de suporte

Sobre este guia

Copyright © G.D.G. Software 2013-2026

Todos os direitos reservados. Nenhuma parte desta obra pode ser reproduzida de qualquer forma ou por qualquer meio, gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, registro em fita ou sistemas de armazenamento e recuperação de informações, sem a permissão por escrito do editor.

É concedida aos clientes licenciados do XLS Padlock a permissão para imprimir este guia para uso privado ou educacional.

Microsoft Excel® e Office® são marcas registradas da Microsoft Corporation.

Quaisquer produtos mencionados neste documento podem ser marcas comerciais e/ou marcas registradas de seus respectivos proprietários. O editor e o autor não reivindicam qualquer direito sobre essas marcas comerciais.

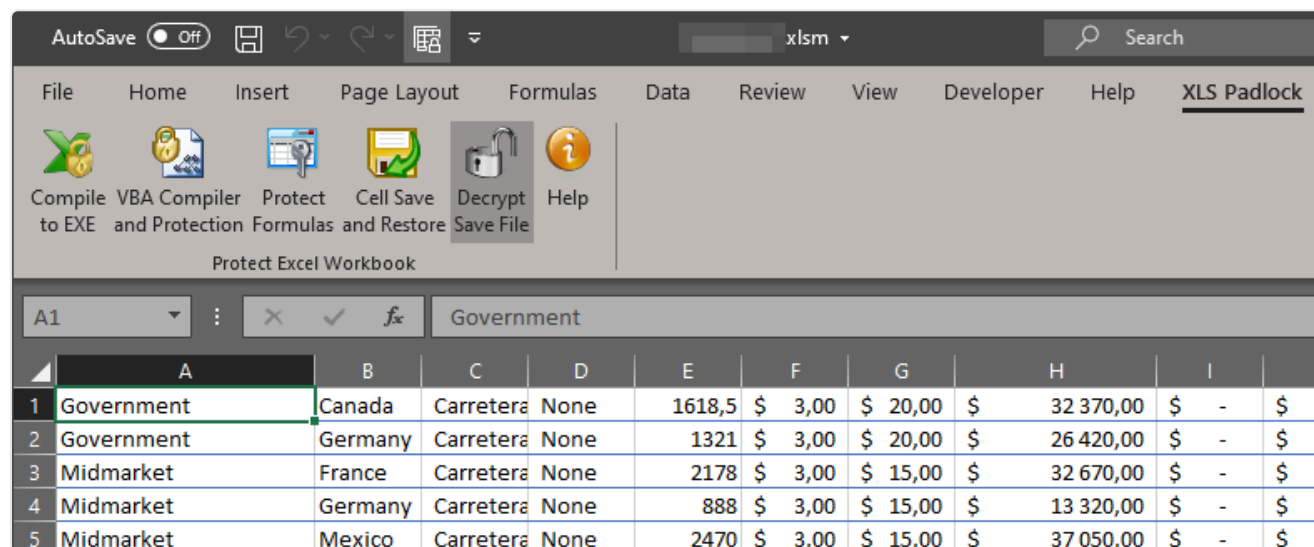
Embora todas as precauções tenham sido tomadas na preparação deste documento, o editor e o autor não assumem qualquer responsabilidade por erros ou omissões, nem por danos resultantes do uso das informações aqui contidas ou do uso dos programas e do código-fonte que o acompanham. Em nenhuma hipótese o editor ou o autor serão responsáveis por qualquer perda de lucros ou qualquer outro dano comercial causado ou supostamente causado direta ou indiretamente por este documento.

Primeira edição: janeiro de 2013. Atualizado: junho de 2026.

Version 2026.0, [Leia o registro de alterações](#)

Introdução ao XLS Padlock

O XLS Padlock é **uma poderosa solução de proteção e licenciamento para o Microsoft Excel**, projetada para proteger suas planilhas contra cópia. Ele funciona como um compilador, permitindo converter suas planilhas do Excel em aplicativos autônomos e seguros.



Isso permite distribuir seus arquivos do Excel com segurança, controlar o acesso dos usuários e impedir a cópia de planilhas. Recursos importantes do Excel podem ser restringidos, incluindo:

- Menu de contexto da planilha (clique com o botão direito)
- Funcionalidade de copiar e colar
- A barra de fórmulas
- Salvar e imprimir
- Acesso ao projeto VBA

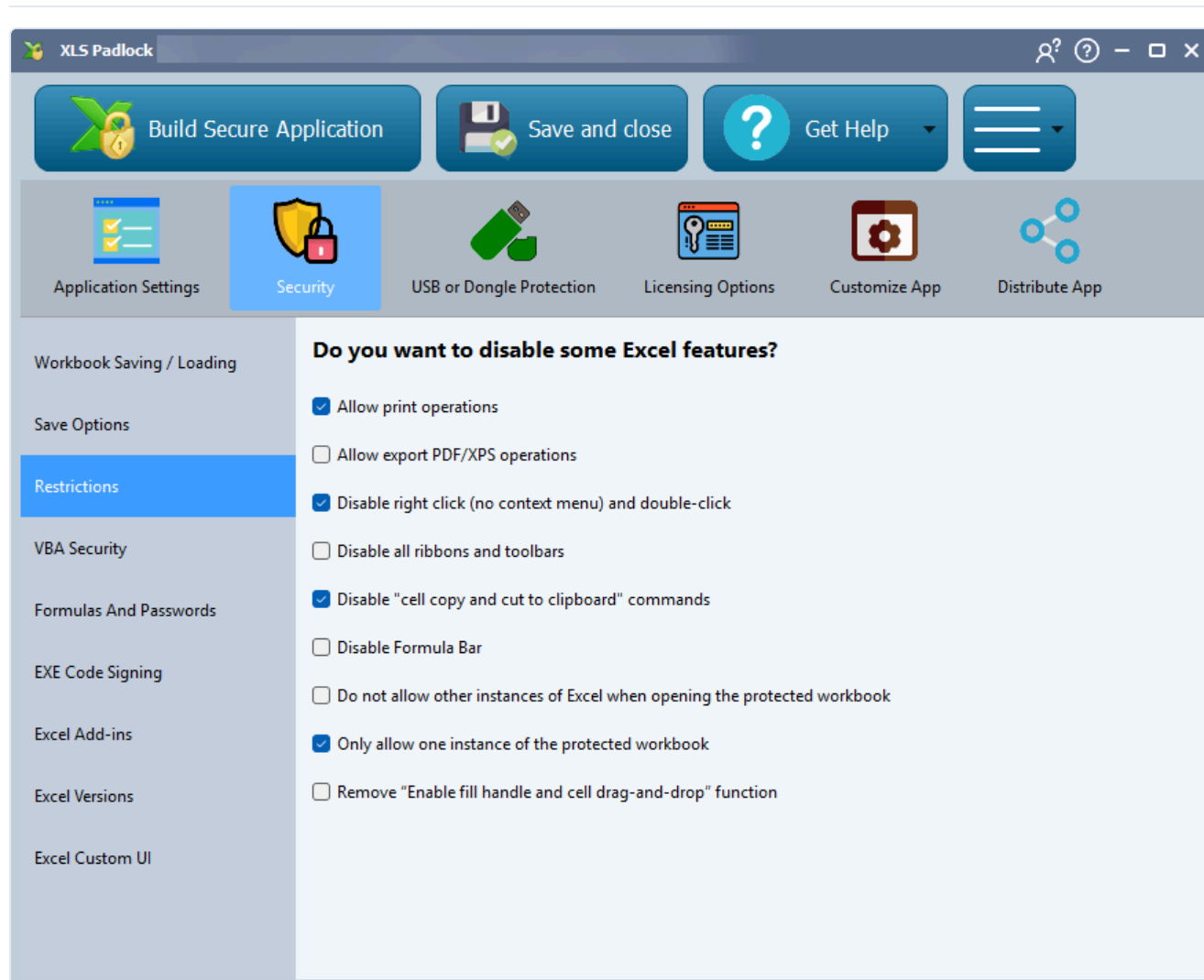
O XLS Padlock oferece várias maneiras de proteger seus arquivos, como **bloqueá-los a um pendrive USB específico**, usar uma **forte proteção por dongle** ou emitir **chaves de ativação bloqueadas por hardware**. Você também obtém controle granular sobre a proteção de fórmulas, permitindo que os usuários utilizem suas fórmulas sem poder visualizá-las ou copiá-las.

Proteção avançada de código VBA

Com seu VBA Compiler integrado, o XLS Padlock permite **compilar código VBA sensível em byte-code seguro, tornando-o inacessível aos usuários finais**. Suas macros VBA ficam realmente protegidas porque o código-fonte original é removido.

O compilador **não é um simples ofuscador**; ele transforma o código VBA em código binário e o armazena com segurança dentro do EXE do aplicativo. Para completar a proteção, você pode bloquear seu projeto VBA, **tornando inúteis as ferramentas de quebra de senha**. [Saiba mais sobre a proteção de código VBA](#).

Opções de segurança abrangentes



Os aplicativos podem ser configurados para **expirar após um determinado número de dias**, um número específico de usos ou em uma data fixa. Uma tela de aviso (nag screen) personalizável pode ser exibida para as versões de avaliação. A **ativação on-line** automatiza o processo de licenciamento, dando a você **controle remoto** sobre quem pode acessar suas planilhas.

Os aplicativos de planilhas protegidas são autônomos e exigem apenas o Microsoft Excel para funcionar. Todos os recursos do Excel são totalmente compatíveis.

👉 Você também pode personalizar seus aplicativos e aplicar sua marca a eles:

- Adicionar um ícone personalizado e informações de copyright.
- Traduzir para qualquer idioma todo o texto visível ao usuário.
- Exibir um contrato de licença (EULA) ou uma tela de abertura na inicialização.
- Assinar digitalmente suas planilhas compiladas com Authenticode.
- Criar um instalador profissional para distribuição.
- Notificar os usuários sobre novas versões com o recurso integrado de atualização pela web.

Compatível com o Office 365 e o Excel 2024, 2021, 2019, 2016, 2013, 2010, 2007 (SP3), 2002, nas versões de 32 bits e 64 bits.

Aviso

Uma versão de 64 bits do Windows é necessária para criar aplicativos de 64 bits.

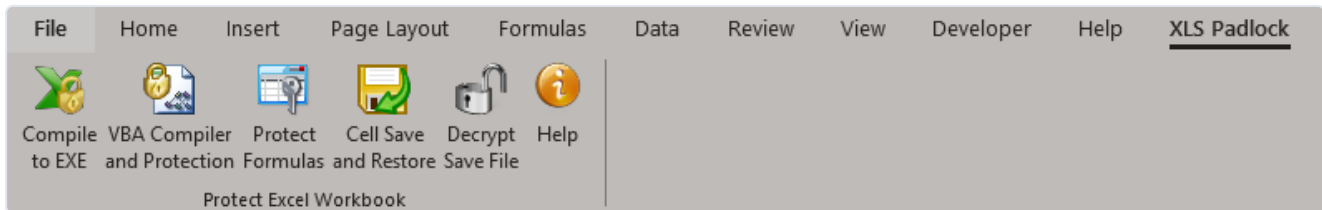
👉 Explore agora:

- [Baixar e instalar o XLS Padlock](#)
- [Como proteger uma planilha do Excel](#)
- [Site do XLS Padlock](#)
- [Novidades](#)

Baixar e instalar

Em nosso site, em <https://www.xlspadlock.com/download>, você deve selecionar o instalador que corresponde à sua versão do Microsoft Excel (32 bits ou 64 bits). O instalador verificará se você está instalando a versão correta.

Após uma instalação bem-sucedida, o **XLS Padlock integra-se diretamente ao Excel**, aparecendo como uma nova guia na faixa de opções. Para usar o XLS Padlock, primeiro você deve abrir sua planilha do Excel.

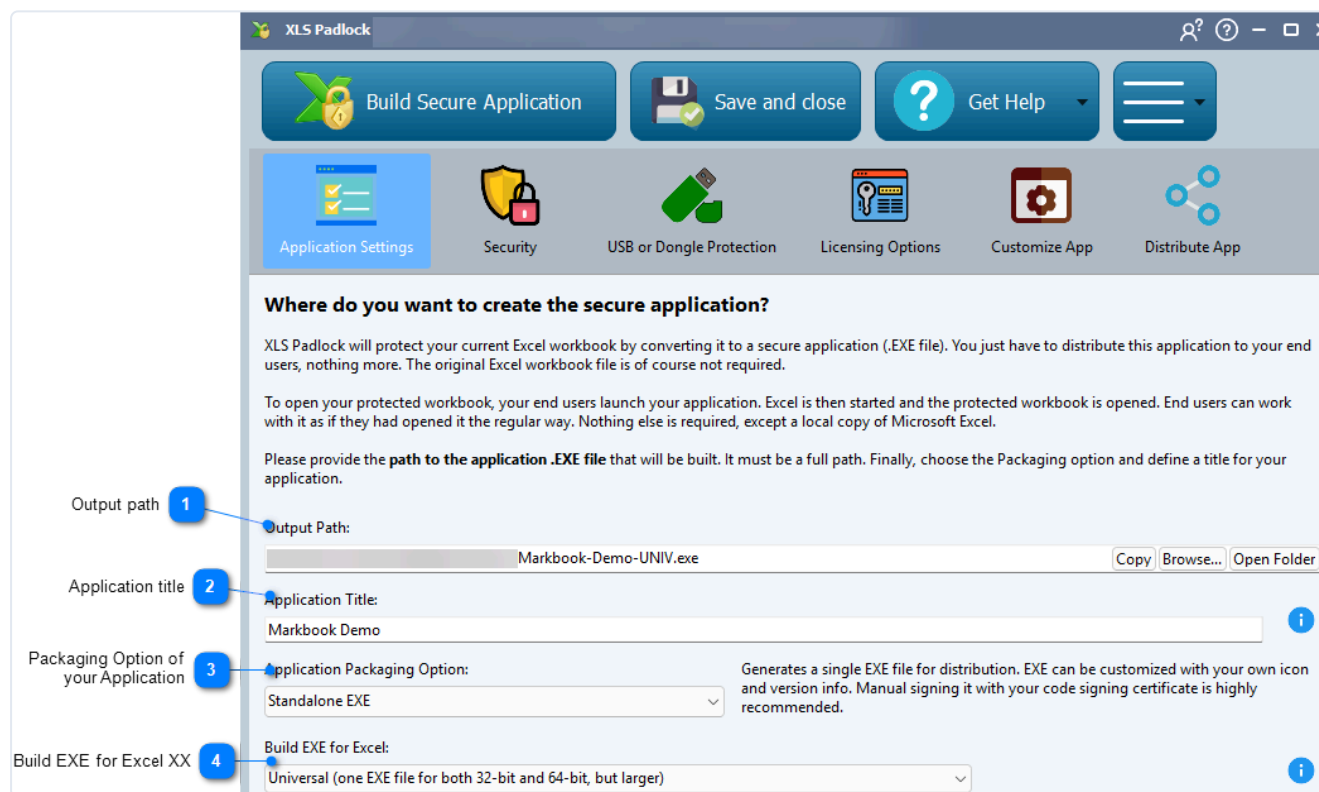


Também são criados dois atalhos na sua área de trabalho: um para este guia do usuário e outro para o XLS Padlock Manager.

- [XLS Padlock Manager](#)
- [Como proteger uma planilha do Excel](#)

Configurações do aplicativo

A página Application Settings (configurações da aplicação) contém as configurações obrigatórias do seu projeto.



Configurações principais

Output Path

👉 Define onde o arquivo EXE final será criado.

O XLS Padlock compila sua planilha em um único arquivo executável (EXE). No campo "Output Path" (caminho de saída), você deve especificar o caminho completo e o nome do arquivo onde deseja que sua aplicação seja criada.

DICA

Não se esqueça de incluir a extensão `.exe` no nome do arquivo.

Também são aceitos caminhos relativos; eles serão relativos à pasta que contém o arquivo da planilha de origem. Por padrão, o XLS Padlock sugere um caminho na mesma pasta que a planilha de origem, mas você pode alterá-lo para qualquer local.

Application Title

👉 Define o título da janela principal da sua aplicação.

Você pode definir um título personalizado para sua aplicação. Esse título é exibido na barra de título da janela, substituindo o texto padrão "Microsoft Excel".

Se você quiser exibir dinamicamente o nome de um arquivo de salvamento carregado, pode utilizar os seguintes marcadores de posição no título:

- `%SAVEFILENAME%` : substituído apenas pelo nome do arquivo (por exemplo, `MyData.xlsx`).
- `%SAVEFULLNAME%` : substituído pelo caminho completo até o arquivo de salvamento (por exemplo, `C:\Users\Me\Documents\MyData.xlsx`).

Por exemplo, um título definido como `My Application, %SAVEFILENAME%` seria exibido como "My Application, MyData.xlsx" quando o usuário tiver esse arquivo aberto.

OBSERVAÇÃO

Se a planilha original, ainda não salva, estiver carregada, os marcadores de posição ficarão vazios.

Application Packaging Option

Consulte [Opções de empacotamento](#): escolha entre um EXE independente ou um pacote para distribuição.

Build EXE for Excel

Consulte [Build EXE for Excel](#): especifique se deseja compilar para 32 bits, 64 bits ou compatibilidade universal.

Configuração adicional

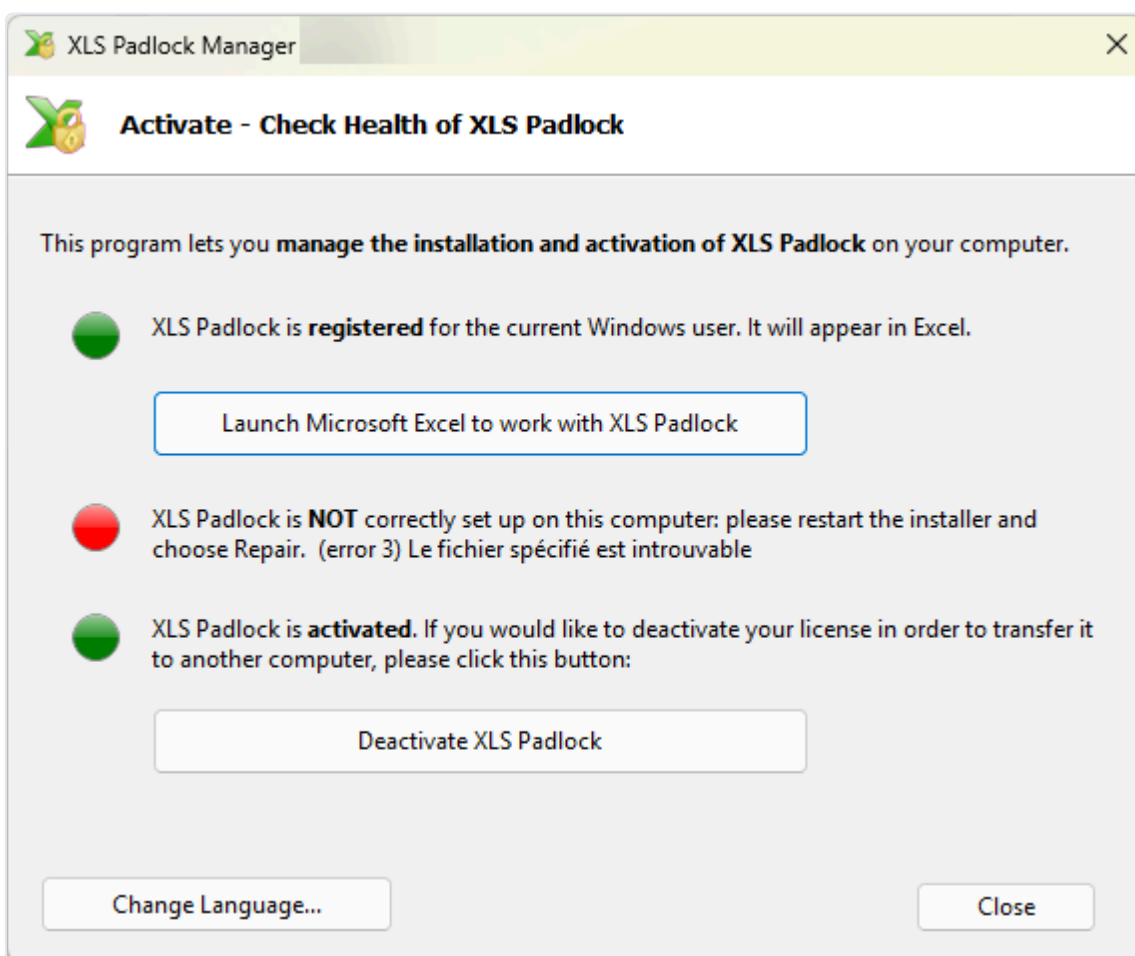
- [Configure Advanced Options](#)
- [Add Companion Files](#)

XLS Padlock Manager

O XLS Padlock Manager é um aplicativo independente que permite **verificar se o XLS Padlock está corretamente configurado e registrado para o usuário atual do Windows**.

O XLS Padlock **deve ser registrado para cada usuário do Windows que queira utilizá-lo**. Quando você instala o XLS Padlock em um computador, normalmente ele fica registrado apenas para a conta de administrador que realizou a instalação. As demais contas de usuário devem usar o XLS Padlock Manager para registrar o software.

- Para registrar o XLS Padlock na sua conta de usuário do Windows, inicie o "XLS Padlock for Excel, Manager" pela área de trabalho do Windows. A seguinte janela será exibida:



- Se todos os indicadores estiverem em verde, sua instalação está correta.
- Se o primeiro indicador estiver em vermelho, clique em "**Enable XLS Padlock for the current Windows user**" (Habilitar o XLS Padlock para o usuário atual do Windows). O XLS Padlock será então registrado, e você poderá começar a usá-lo. Esta ação não requer direitos administrativos.



OBSERVAÇÃO

Se você tiver uma licença do XLS Padlock, também poderá usar o manager para ativá-la.

👉 [Como proteger uma planilha do Excel](#)

Como proteger uma planilha do Excel

Tutorial em vídeo disponível

Confira estes tutoriais em vídeo sobre o XLS Padlock com um especialista em Excel:

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

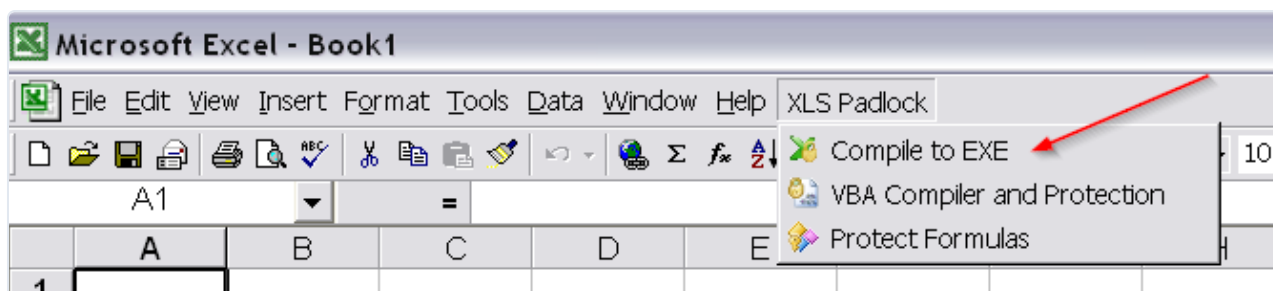
1\ Abra e salve sua planilha

Comece abrindo no Excel a planilha que você deseja proteger. Certifique-se de que todas as alterações estejam salvas antes de continuar.

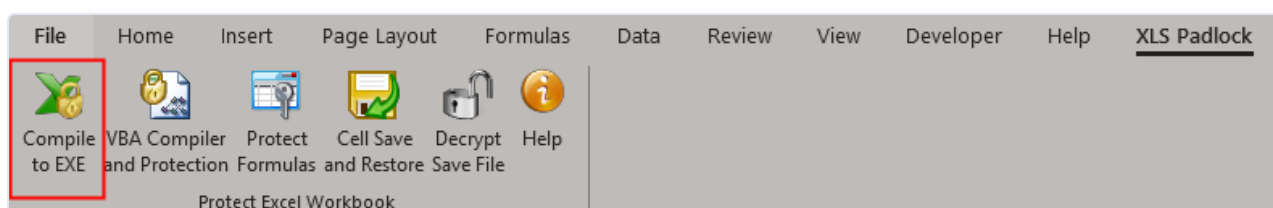
2. Configure as opções de proteção

Para começar, abra a interface do XLS Padlock a partir da faixa de opções do Excel:

- **Antes do Excel 2007:** use o menu "XLS Padlock".



- **Excel 2007 e posteriores:** clique em "Secure Compile to Exe" na guia "XLS Padlock" da faixa de opções.



OBSERVAÇÃO

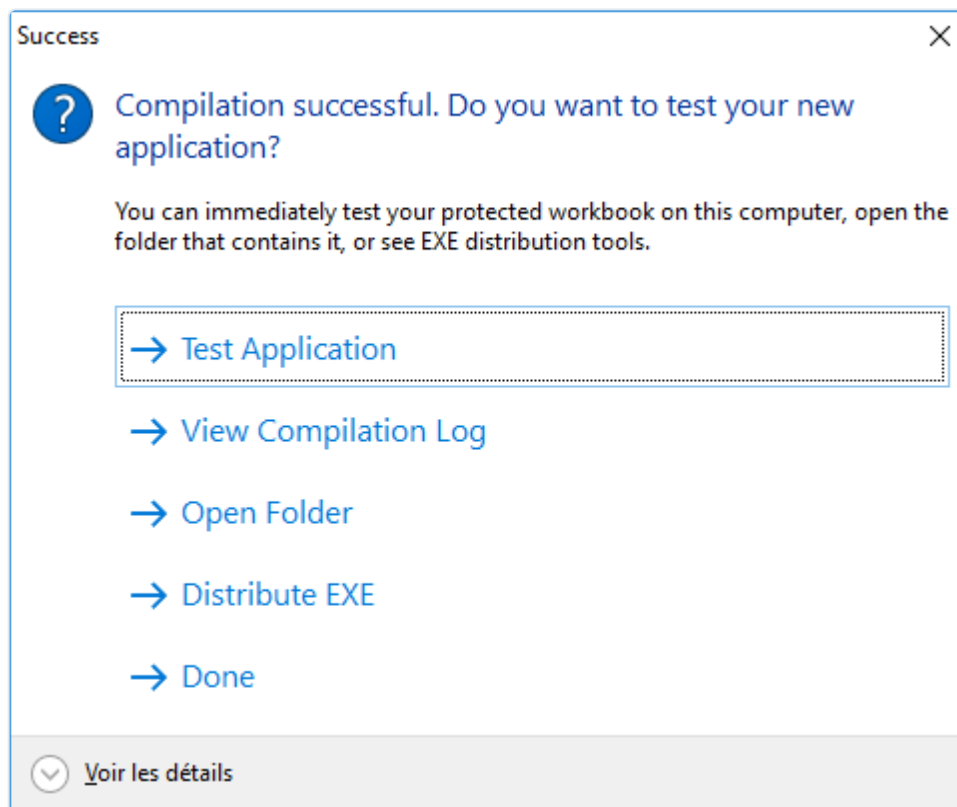
Sempre salve sua planilha do Excel antes de abrir a interface do XLS Padlock para garantir que todas as alterações recentes sejam incluídas na compilação.

3. Crie sua aplicação

Após configurar os ajustes da sua aplicação, clique no botão "Build Secure Application" ou pressione **F5**. O XLS Padlock compilará sua planilha e criará o arquivo EXE da aplicação.



Ao finalizar, será exibida uma mensagem de confirmação:



A partir desta caixa de diálogo, você pode executar imediatamente a aplicação, visualizar o registro de compilação (especialmente útil para solucionar erros) ou abrir a pasta de destino.

DICA

O XLS Padlock não modifica o arquivo da sua planilha original do Excel durante a compilação. Mantenha sempre um backup da sua planilha original.

Solução de problemas de formatação

Se sua planilha protegida exibir resultados estranhos ou problemas de formatação, tente ativar a opção "Use Excel automation for formula protection" na página "Formulas and Passwords".

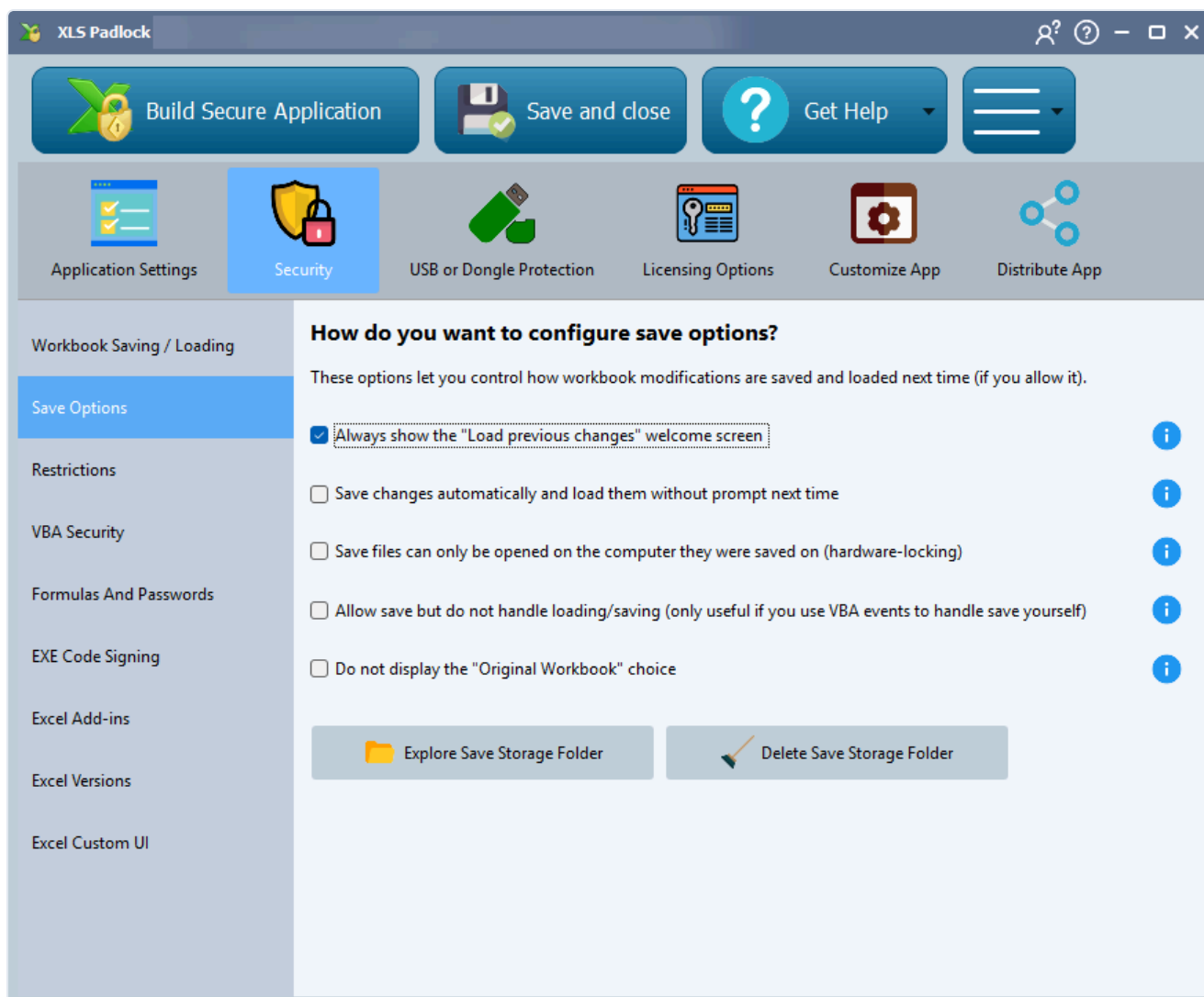
👉 Consulte também:

[Como distribuir uma planilha protegida](#)

Segurança

O XLS Padlock oferece muitas formas de proteger sua planilha do Excel, incluindo opções de salvamento, restrições de recursos e condições para abrir seu aplicativo, como exigir uma [chave de ativação](#) ou a [presença de um dongle ou pendrive USB](#).

Este tópico apresenta uma visão geral das opções de segurança disponíveis em "**Security**" (Segurança) no XLS Padlock.



Salvamento / carregamento da planilha

Configure se os usuários finais podem salvar as alterações na planilha protegida. Quando o salvamento está habilitado, o XLS Padlock cria arquivos seguros `.XLSC` ou `.XLSCE` que só podem ser abertos pelo seu aplicativo, protegendo os dados do usuário.

➔ [Saiba mais sobre o salvamento e o carregamento da planilha](#)

Opções de salvamento

Ajuste com precisão o comportamento de salvamento com opções como salvar automaticamente as alterações ao fechar, vincular por hardware (bloqueio por hardware) os arquivos salvos a um computador específico ou exibir sempre a tela de boas-vindas para que os usuários escolham entre começar do zero e carregar um trabalho anterior.

→ [Explore as opções avançadas de salvamento](#)

Restrições

Restrinja recursos importantes do Excel para reforçar a segurança. Você pode desabilitar o menu de contexto do botão direito, a barra de fórmulas, os comandos de copiar e colar, a impressão e muito mais para impedir que os usuários acessem ou modifiquem partes sensíveis da sua planilha.

→ [Veja os recursos de restrição](#)

Segurança do VBA

Proteja seu código VBA contra visualização ou modificação. Você pode bloquear o projeto VBA com uma senha que fica oculta para o usuário ou bloquear completamente o acesso ao editor de VBA (VBE), tornando inúteis as ferramentas de quebra de senhas. Para a máxima proteção, utilize o VBA Compiler.

→ [Veja os diferentes meios de proteção do VBA](#)

Fórmulas e senhas

Proteja suas fórmulas e a estrutura das planilhas. Use a proteção de fórmulas do XLS Padlock para permitir que os usuários interajam com as fórmulas sem poder vê-las ou copiá-las. Você também pode gerenciar as senhas da planilha e das folhas de cálculo diretamente pela interface.

→ [Saiba mais sobre a proteção de fórmulas](#)

Assinatura de código do EXE

Assine digitalmente seu arquivo `.EXE` compilado para gerar confiança entre seus usuários e evitar avisos do Windows SmartScreen e de softwares antivírus. O XLS Padlock pode automatizar o processo de assinatura se você tiver um certificado de assinatura de código.

→ [Como assinar digitalmente o código do seu aplicativo](#)

Suplementos do Excel

Por padrão, o XLS Padlock desabilita a maioria dos suplementos por segurança. Esta seção permite que você reabilite seletivamente determinados suplementos COM ou suplementos comuns do Excel (como o Analysis ToolPak ou o Solver) que sua planilha precisa para funcionar corretamente.

→ [Como gerenciar os suplementos do Excel](#)

Versões do Excel

Especifique quais versões do Microsoft Excel são compatíveis com seu aplicativo protegido. Se um usuário tentar executar o EXE em uma versão não compatível, uma mensagem de erro personalizável

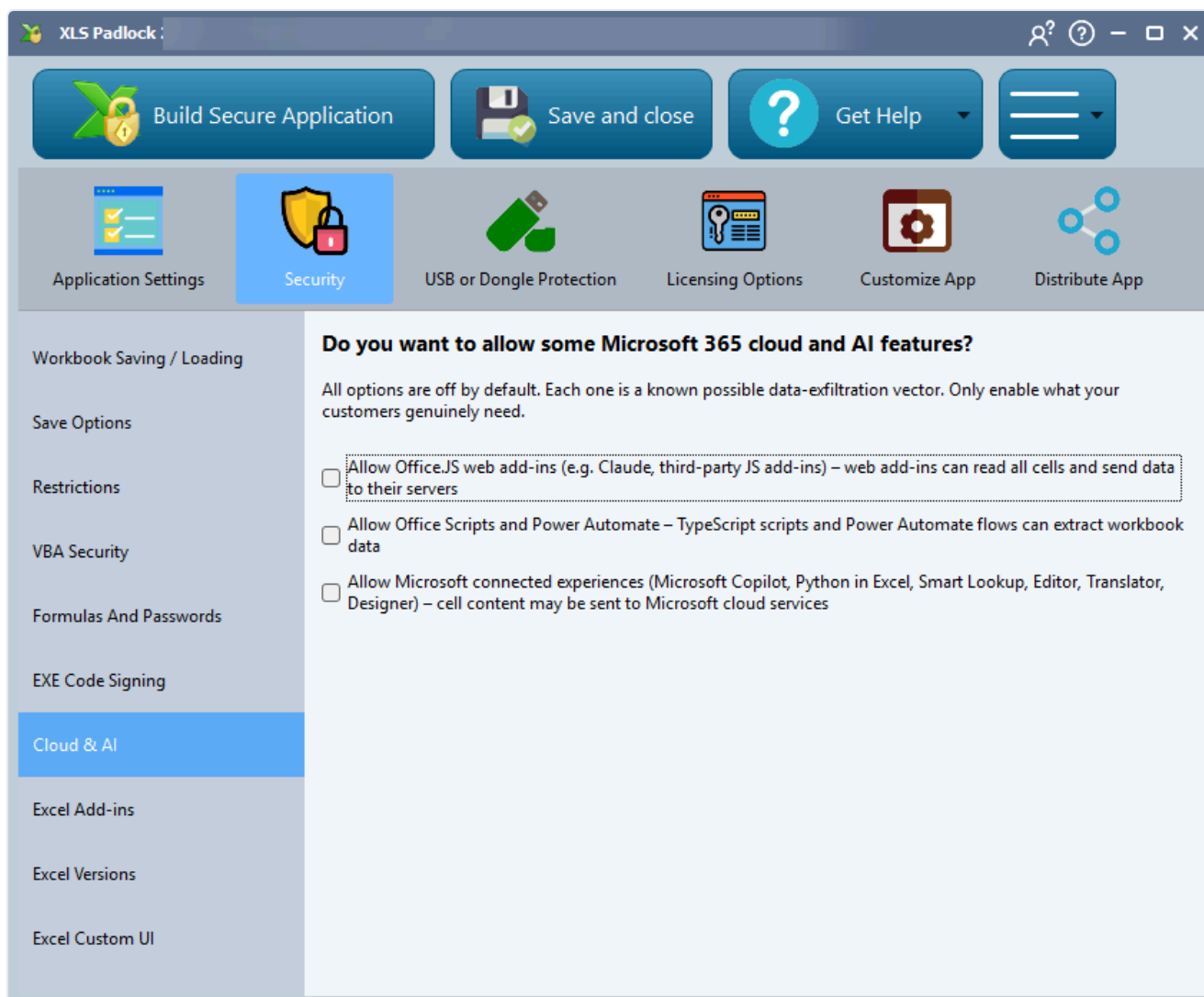
será exibida.

[→ Como exigir versões específicas do Excel](#)

Opções de nuvem e IA

As versões modernas do Excel incluem vários recursos de nuvem e IA que podem ler a sua planilha e enviar o conteúdo dela para servidores externos. Para manter a sua aplicação protegida selada, o XLS Padlock bloqueia todos eles por padrão. Você decide, um por um, quais permitir.

Você encontrará essas configurações no designer do XLS Padlock, na aba **Security** (Segurança), na página **Cloud & AI** (Nuvem e IA).



Por que esses recursos são bloqueados por padrão

Cada opção desta página é uma via conhecida para que os dados da planilha saiam da máquina do usuário final. Uma planilha protegida tem como objetivo manter as suas fórmulas, dados e código VBA sob o seu controle, então o XLS Padlock parte da posição mais segura: tudo desativado. Ative apenas o que os seus clientes realmente precisam.

Opções disponíveis

As três opções vêm desmarcadas (bloqueadas) por padrão.

Allow Office JS web add-ins

Os complementos web do Office JS (também chamados de web add-ins) são executados dentro do Excel e podem ler todas as células da planilha aberta e enviar esses dados para os próprios servidores deles. Isso inclui complementos de terceiros, como o plugin **Claude for Excel**, e qualquer outro complemento JS instalado a partir da Office store ou carregado de forma lateral (sideloaded).

Quando esta opção está desativada, os complementos web não podem ser carregados contra a sua planilha protegida. Habilite-a apenas se a sua aplicação depender de um complemento web específico no qual você confia.

Allow Office Scripts and Power Automate

Os Office Scripts (scripts de automação em TypeScript) e os fluxos do Power Automate podem ler o conteúdo da planilha e retirá-lo da aplicação. Deixe esta opção desativada, a menos que o seu fluxo de trabalho dependa deles.

Allow Microsoft connected experiences

As experiências conectadas da Microsoft (Microsoft connected experiences) enviam o conteúdo das células para os serviços de nuvem da Microsoft. Isso abrange Microsoft Copilot, Python in Excel, Smart Lookup, Editor, Translator e Designer. Habilite-a apenas se os seus clientes precisarem desses recursos e você aceitar que o conteúdo de células correspondente poderá ser processado na nuvem da Microsoft.

Observações

- Essas opções se aplicam à aplicação compilada, não ao designer do XLS Padlock.
- Bloquear essas superfícies também oculta os pontos de acesso delas na faixa de opções do Excel (por exemplo, o menu suspenso Add-ins, a aba Automate e o botão Copilot), de modo que os usuários finais não são convidados a usar um recurso que está desativado.
- Os complementos COM e os complementos integrados do Excel são gerenciados separadamente. Consulte [Como gerenciar complementos do Excel](#).

Melhorar a proteção

Por que compilar para um EXE não é suficiente

Ao compilar uma planilha em um aplicativo protegido, o arquivo de planilha original (XLSX, XLSM, etc.) é criptografado e incorporado ao arquivo .EXE resultante. Isso impede que os usuários acessem diretamente o arquivo original.

A proteção básica não é suficiente

No entanto, apenas compilar a sua planilha fornece somente uma camada básica de segurança. Pense nisso como fechar uma porta sem trancá-la. Para proteger de verdade a sua propriedade intelectual, você deve habilitar os recursos de proteção avançada oferecidos pelo XLS Padlock.

Para funcionar, o XLS Padlock precisa carregar a sua planilha na memória do Excel. **Sem ativar as camadas de segurança adicionais do XLS Padlock**, um invasor sofisticado poderia, potencialmente, extrair a planilha original da memória enquanto o aplicativo está em execução. Embora essa não seja uma tarefa trivial, ela evidencia a necessidade de uma proteção mais forte.

Torne uma planilha extraída inútil

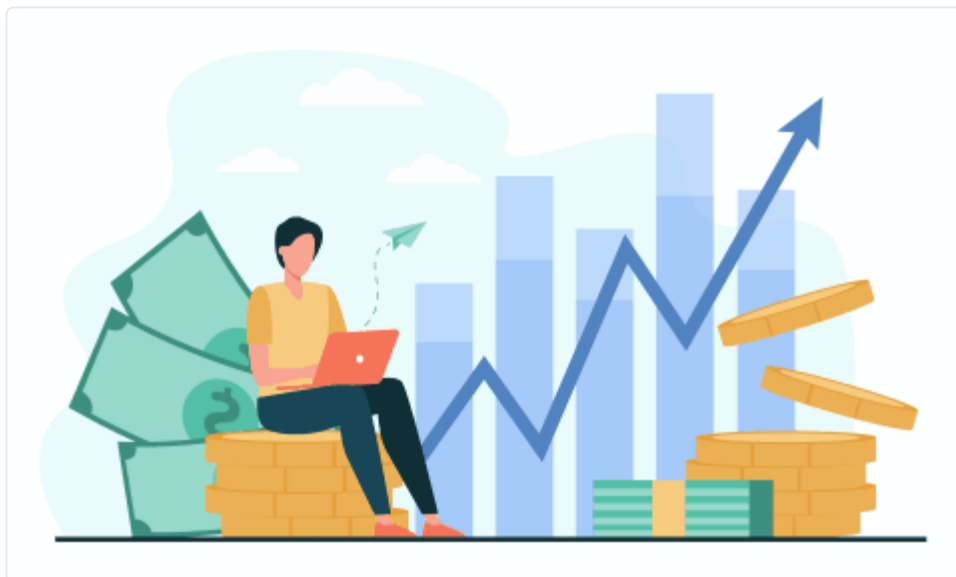
Quando você aplica corretamente os recursos de segurança do XLS Padlock, a planilha extraída torna-se inútil. Mesmo que alguém extraia o arquivo, ele não funcionará corretamente. A proteção é especialmente forte para planilhas que contêm fórmulas e/ou código VBA, pois esses componentes críticos são criptografados ou compilados, impedindo de forma eficaz o uso não autorizado ou a adulteração.

Recursos de segurança mais eficazes

Recomendamos habilitar os seguintes recursos para obter a proteção mais forte:

- [Proteja suas fórmulas com a proteção de fórmulas](#)
- [Proteja seu VBA com o VBA Compiler](#)
- [Evite ataques comuns a VBA e OLE](#)
- [Proíba o acesso ao editor de VBA \(VBE\)](#)
- [Proteja sua planilha com senha](#)
- [Desative complementos do Excel desnecessários](#)

Controle de acesso à planilha



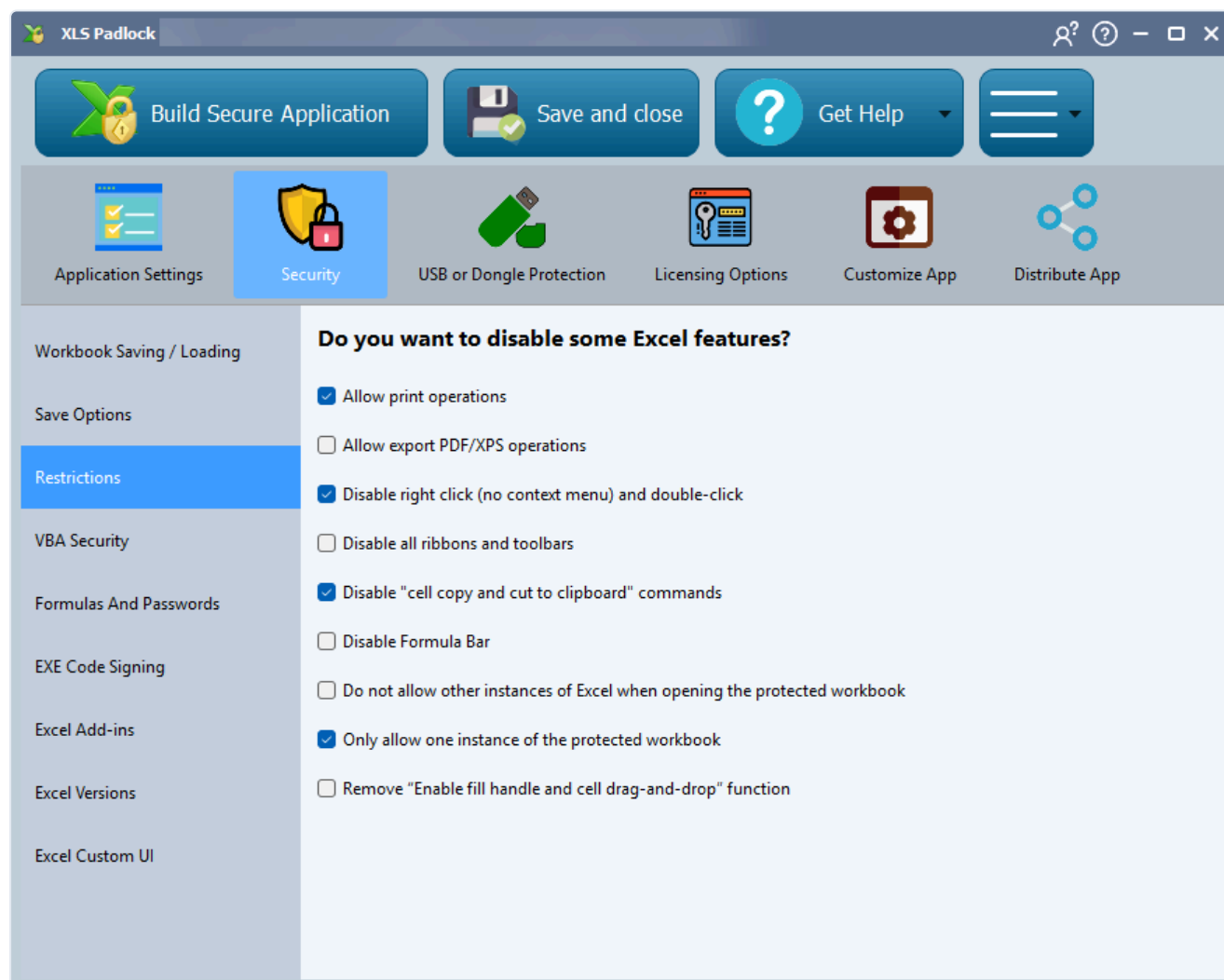
O XLS Padlock oferece diversos recursos de segurança para **restringir o acesso às suas planilhas do Excel apenas a usuários autorizados**. Você pode gerar chaves de ativação, definir versões de avaliação com tempo limitado e muito mais. Um objetivo principal do XLS Padlock é permitir que você **venda licenças para suas planilhas**, oferecendo funcionalidades para reduzir a pirataria e o compartilhamento não autorizado.

Este tópico apresenta uma visão geral dos recursos de segurança dedicados ao controle de acesso à planilha:

- **Controle quem pode usar sua planilha:** [Configure chaves de ativação](#).
- **Evite o compartilhamento vinculando uma licença a um único computador:** Use [chaves de ativação com bloqueio por hardware](#).
- **Gerencie ativações remotamente e desative o acesso:** Use [ativação online](#), [desativação](#) e [validação](#).
- **Crie versões de avaliação da sua planilha:** Saiba [como criar planilhas de avaliação](#).
- **Restrinja as ações do usuário:** [Proíba impressão](#), [exportação](#), [clique com o botão direito](#) e [muito mais](#).
- **Exija um dispositivo físico para o acesso:** Vincule o aplicativo a um [dongle ou pen drive USB](#).
- **Venda licenças e assinaturas online:** Integre-se a plataformas de comércio eletrônico como [WooCommerce](#) ou [FastSpring](#).

Restrições da planilha do Excel

Este tópico aborda as opções disponíveis na seção 'Security > Restrictions' do XLS Padlock, que permitem restringir certas ações do usuário na sua planilha protegida.



Allow Print Operations

Para impedir que os usuários imprimam sua planilha, desative a opção **Allow print operations** (permitir operações de impressão) nas configurações de [Security](#).

Se essa opção estiver desativada, qualquer tentativa de impressão será bloqueada e o usuário verá a mensagem de erro "Printing is not allowed".

Allow Export to PDF/XPS

Para impedir que os usuários exportem o conteúdo da planilha para arquivos PDF ou XPS, desative a opção "Allow export PDF/XPS operations" (permitir operações de exportação para PDF/XPS) nas configurações de [Security](#).

Disable Right-Click (and optionally Double-Click)

Para desabilitar o menu de contexto do botão direito do mouse nas células da sua planilha, ative a opção "Disable right-click" (desabilitar clique com o botão direito) nas configurações de [Security](#).

Isso costuma ser usado para impedir que os usuários copiem, coleem ou modifiquem os dados e a formatação das células.

Disable All Ribbons and Toolbars

A opção **Disable all ribbons and toolbars** (desabilitar todas as faixas de opções e barras de ferramentas) oculta todas as guias internas da Faixa de Opções e a maioria dos comandos do Menu do Microsoft Office, criando uma interface simplificada, semelhante a um quiosque, para o seu aplicativo.

Esse recurso funciona com o **Excel 2007 e versões posteriores**. Você pode aplicá-lo definindo a [versão mínima exigida do Excel](#) para a sua planilha.

Disable "Cell Copy and Cut to Clipboard" Commands

Ative esta opção para desabilitar os comandos **Copy** (copiar) e **Cut** (recortar) das células. Isso impede que os usuários finais copiem com facilidade dados da sua planilha para a área de transferência a fim de reutilizá-los em outros aplicativos.

Disable Formula Bar

A opção **Disable Formula Bar** (desabilitar a barra de fórmulas) oculta a barra de fórmulas do Excel. Além disso, a opção correspondente no menu Exibir do Excel é desabilitada, de modo que os usuários finais não consigam reativá-la.

Para maior segurança, também recomendamos que você [desabilite o acesso ao editor do VBA](#).

Do Not Allow Other Instances of Excel

Quando ativada, esta opção verifica se já existe outra instância do Excel em execução quando um usuário inicia sua planilha compilada. Se um processo do Excel existente for encontrado, uma mensagem solicitará que o usuário o feche antes de prosseguir.

Isso garante que a instância do Excel que executa sua planilha protegida seja a primeira e a única, proporcionando um ambiente mais controlado.

Esta opção pode ser encontrada nas configurações de [Security](#).

Only Allow One Instance of the Protected Workbook

A opção "Only allow one instance of the protected workbook" (permitir apenas uma instância da planilha protegida), encontrada nas configurações de [Security](#), impede que os usuários finais executem mais de uma instância do seu aplicativo protegido ao mesmo tempo. Se um usuário tentar abrir o aplicativo novamente enquanto ele já está em execução, a janela existente será trazida para o primeiro plano. Esse recurso impede de forma eficaz que os usuários abram várias instâncias da mesma planilha protegida.

Nota de compatibilidade

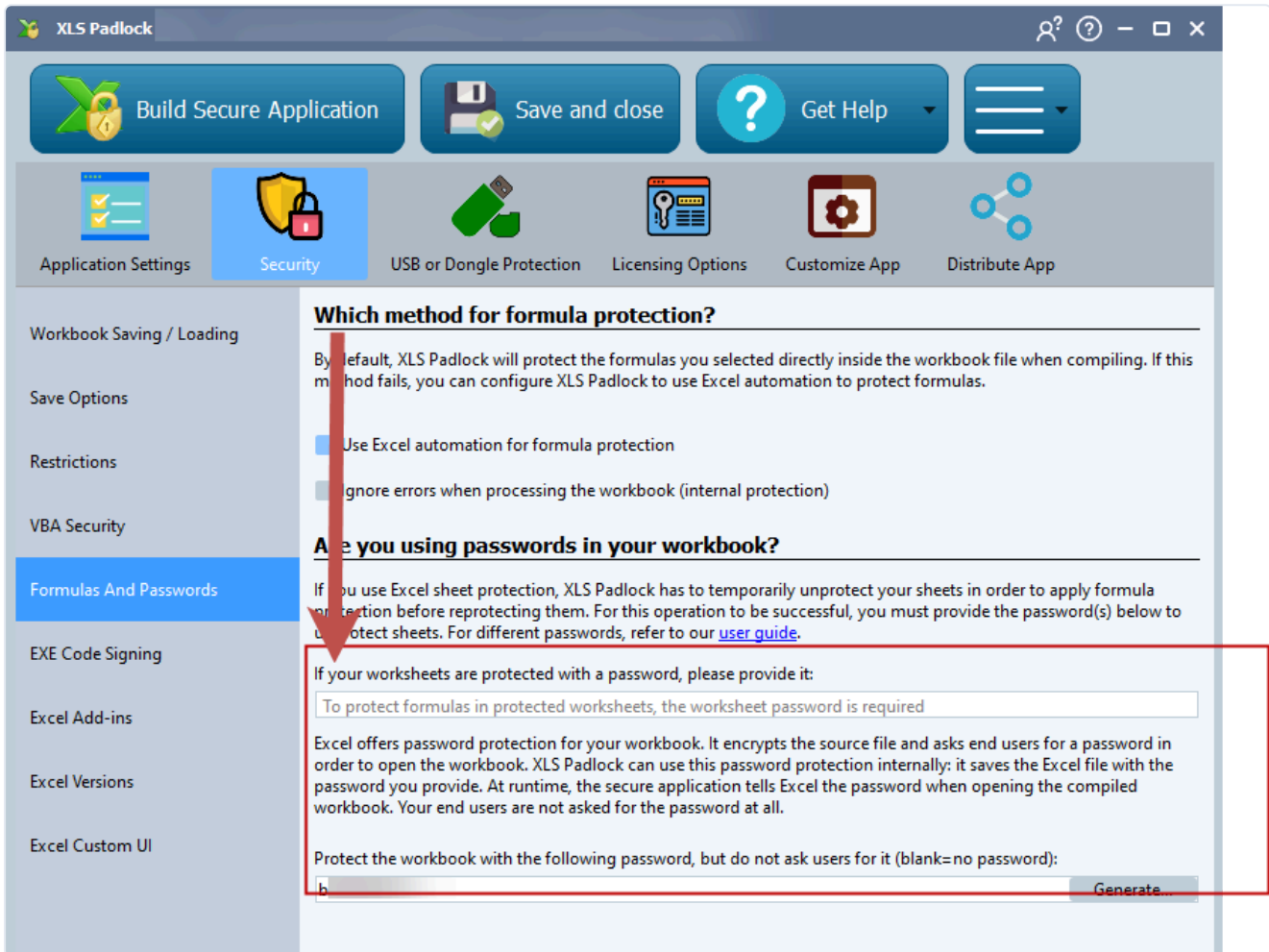
Em versões do Excel anteriores a 2013, será exibida uma caixa de mensagem com o texto "An instance of this application is already running" em vez de focar a janela existente.

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

Esta opção "Remove "Enable fill handle and cell drag-and-drop" function" impede que os usuários finais usem a alça de preenchimento e o arrastar e soltar de células na planilha protegida. É a mesma das opções gerais do Excel ("Review", "Check accessibility", "Options accessibility", "Advanced", "Enable fill handle and cell drag-and-drop"), mas permite substituir a escolha do usuário local.

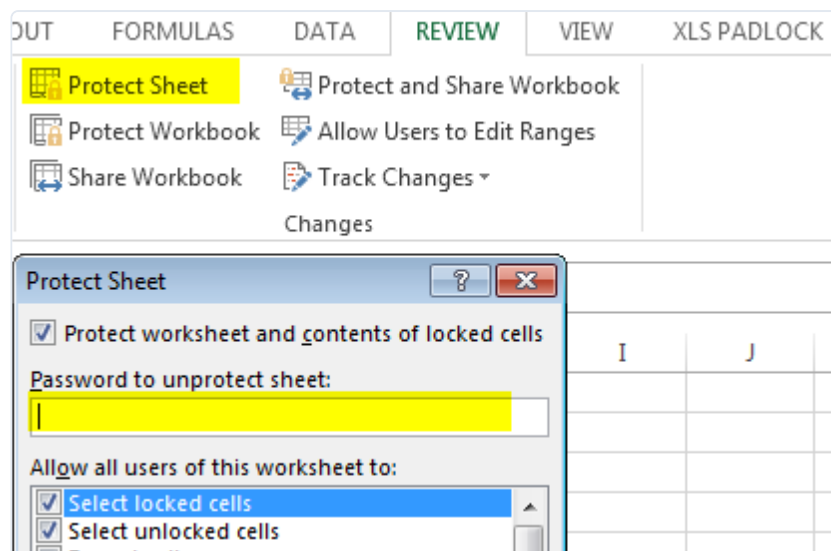
Proteção por senha da planilha

O XLS Padlock permite gerenciar tanto as senhas da planilha quanto as das guias para reforçar a segurança e a funcionalidade do seu aplicativo protegido. Este guia explica como configurar essas opções de senha.



Proteção por senha da guia

Se você usar a proteção de guias integrada do Excel para restringir o acesso às células, o XLS Padlock precisa conhecer sua senha para gerenciar suas próprias camadas de segurança.



Quando a [Proteção de fórmulas](#) do XLS Padlock está ativa, ela precisa desproteger temporariamente suas guias, aplicar sua proteção e, em seguida, protegê-las novamente. Para realizar essa operação com sucesso, você deve fornecer a senha usada para a proteção de guias no Excel.

OBSERVAÇÃO

O XLS Padlock reaplica automaticamente as mesmas opções de proteção de guia que você definiu na planilha original.

Se você usar uma única senha para todas as guias protegidas, digite-a no campo "If your worksheets are protected with a password, please provide it" (se suas guias estiverem protegidas com uma senha, forneça-a).

Se você usar senhas diferentes para guias diferentes, deve fornecê-las usando o seguinte formato JSON:

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

Por exemplo:

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

Proteção silenciosa por senha da planilha

O Excel oferece um recurso para criptografar toda a planilha com uma senha. O XLS Padlock pode gerenciar essa senha para você, fornecendo uma camada adicional de segurança sem solicitar nada aos seus usuários finais.

Quando você define uma senha no campo "Protect the workbook with the following password" (proteger a planilha com a seguinte senha), o XLS Padlock salva a planilha com essa senha. Em tempo de execução, o aplicativo seguro fornece automaticamente a senha ao Excel ao abrir a planilha. **Seus usuários finais nunca recebem a solicitação da senha.**

Essa combinação aumenta a segurança do seu aplicativo ao sobrepor a proteção do XLS Padlock à criptografia nativa do Excel.

Você pode usar o botão **Generate** (gerar) para criar uma senha forte e aleatória.

Não altere a senha após a distribuição

Depois de distribuir um aplicativo protegido com uma senha de planilha específica, não a altere. Se você fizer isso, os usuários finais não conseguirão abrir os dados salvos anteriormente, porque o aplicativo não conseguirá descriptografá-los.

Se você deixar o campo de senha em branco, esse recurso será desabilitado.

Combinar a proteção do Excel e do XLS Padlock

A proteção de fórmulas do XLS Padlock é totalmente compatível com os recursos de proteção de planilhas integrados ao Excel, como células bloqueadas ou ocultas. Quando você utiliza ambos, as células protegidas não exibirão nada na Barra de Fórmulas, e as chamadas subjacentes à função `PLEvalForm` permanecerão 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	$[\ln(S/X) + (r + 0,5 \div 2) \times t] / \div \sqrt{t}$	2.52

Aviso sobre dependências

As fórmulas protegidas que dependem de outras células falharão se essas células forem ocultadas por meio do atributo de formato de célula "Hidden" do Excel. Evite usar o atributo oculto em células que sejam dependências das suas fórmulas protegidas.

Consulte também:

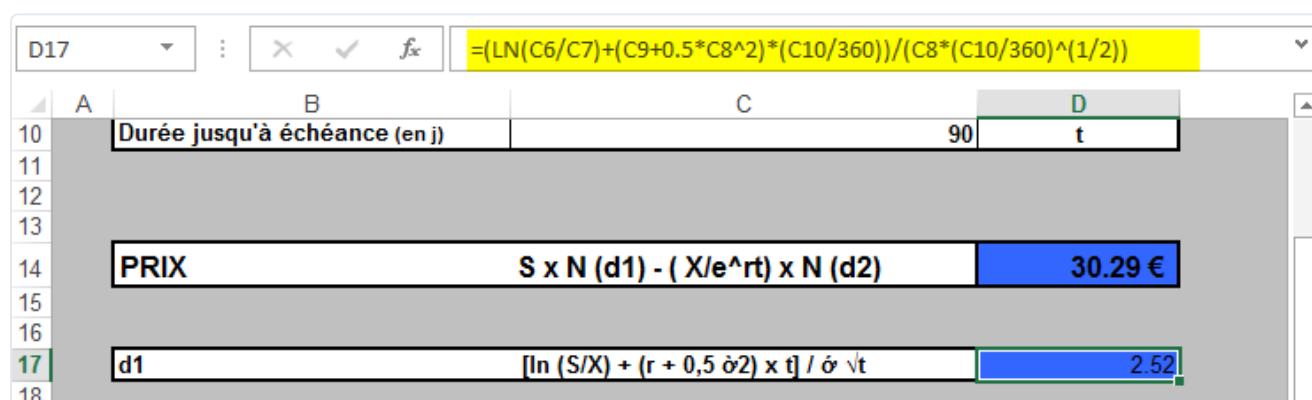
- [Desativar a proteção de fórmulas](#)

Proteção de fórmulas: Excel versus XLS Padlock

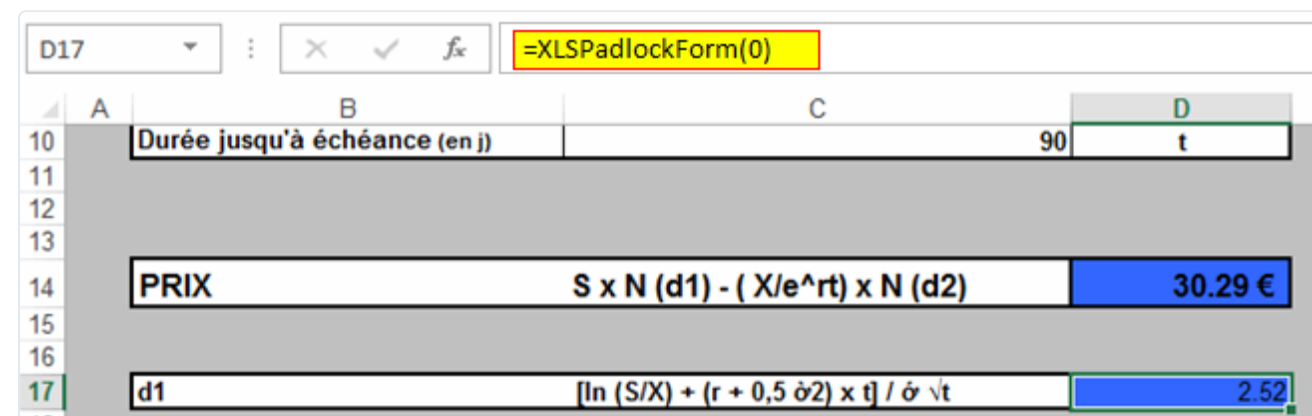
Embora o Excel permita ocultar fórmulas em uma planilha protegida, essa proteção é limitada. O XLS Padlock vai muito além ao substituir suas fórmulas por uma função segura que só fica disponível quando seu aplicativo está em execução.

Na planilha compilada, as fórmulas são substituídas na barra de fórmulas por funções anônimas como `=PLEvalFormD(0;0)` ou `=PLEvalFormD(1;COUNT($F7))`. Mesmo assim, os cálculos continuarão funcionando como esperado.

Antes da proteção, qualquer pessoa pode ver sua fórmula:

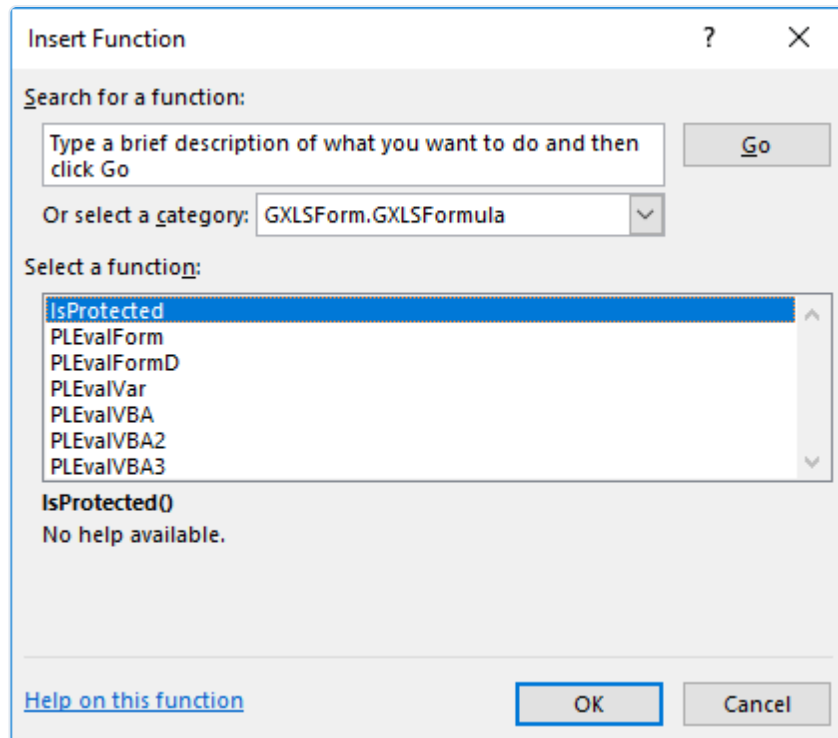


Após a proteção com XLS Padlock, a fórmula é substituída automaticamente, permanecendo funcional:



Isso aumenta consideravelmente a segurança da sua planilha. Como as fórmulas originais são removidas do arquivo da planilha, as células protegidas só podem funcionar corretamente quando a planilha é aberta dentro do aplicativo seguro criado pelo XLS Padlock.

Mesmo em tempo de design, você pode ver que as funções de segurança do XLS Padlock, como `PLEvalForm` e `PLEvalFormD`, são reconhecidas pelo Excel:



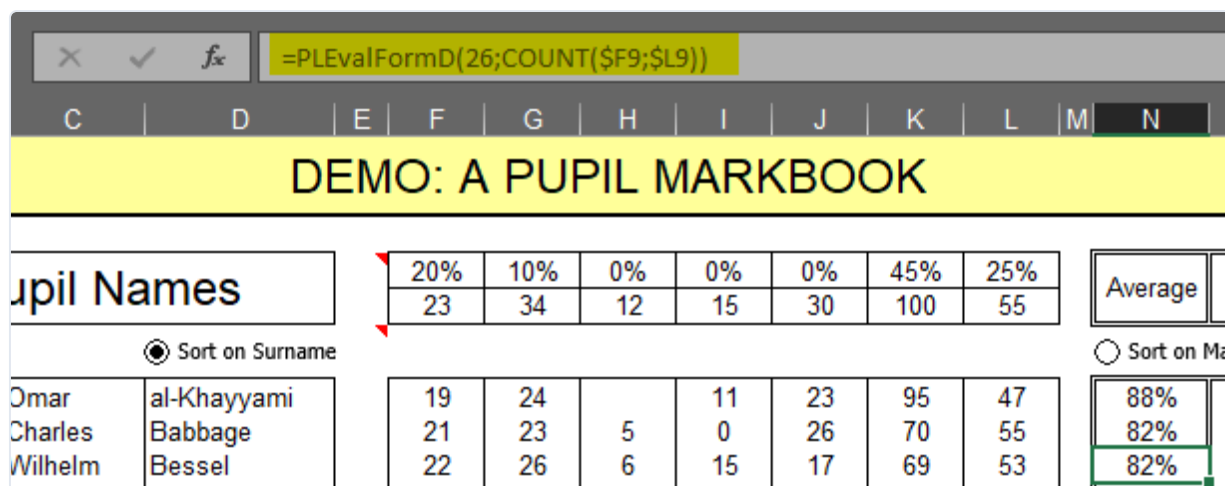
👉 Saiba mais sobre [como proteger células com XLS Padlock](#).

Proteger fórmulas com o XLS Padlock

Além da proteção de células do Excel, **recomendamos fortemente proteger as fórmulas importantes com a própria proteção de fórmulas do XLS Padlock.**

Após a proteção com o XLS Padlock, suas fórmulas não aparecerão mais na barra de fórmulas, **mas continuam totalmente funcionais.** Em vez disso, é exibida uma chamada de função críptica como

PLEvalFormD :

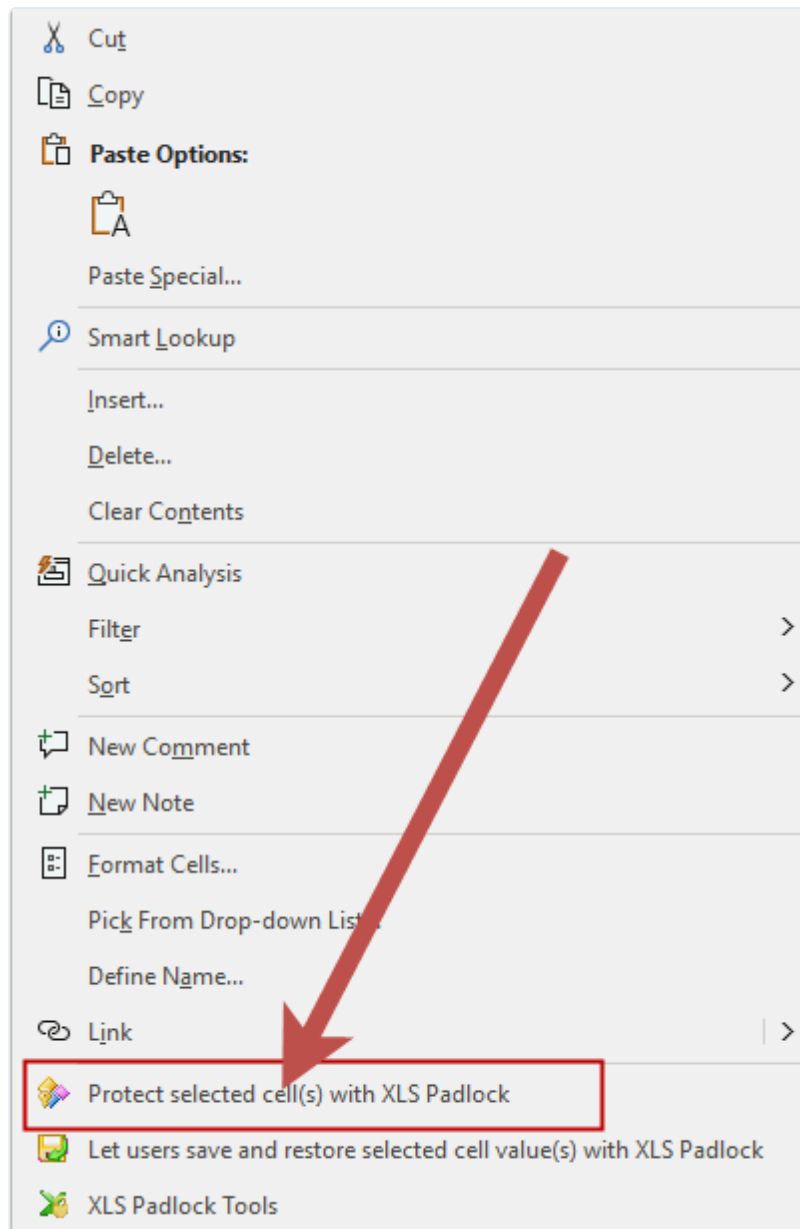


The screenshot shows an Excel spreadsheet titled "DEMO: A PUPIL MARKBOOK". The formula bar displays the protected formula `=PLEvalFormD(26;COUNT($F9;$L9))`. The spreadsheet contains a table of student marks with columns for names and various test scores. The table is sorted by surname.

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

👉 O objetivo é tornar a planilha inutilizável caso alguém consiga [extrair o arquivo da planilha do EXE compilado](#). Como as **fórmulas protegidas pelo XLS Padlock não estão mais no arquivo da planilha**, a planilha extraída não funcionará corretamente.

O XLS Padlock permite que você decida quais células proteger, e você pode **selecionar e proteger várias células de uma vez.**



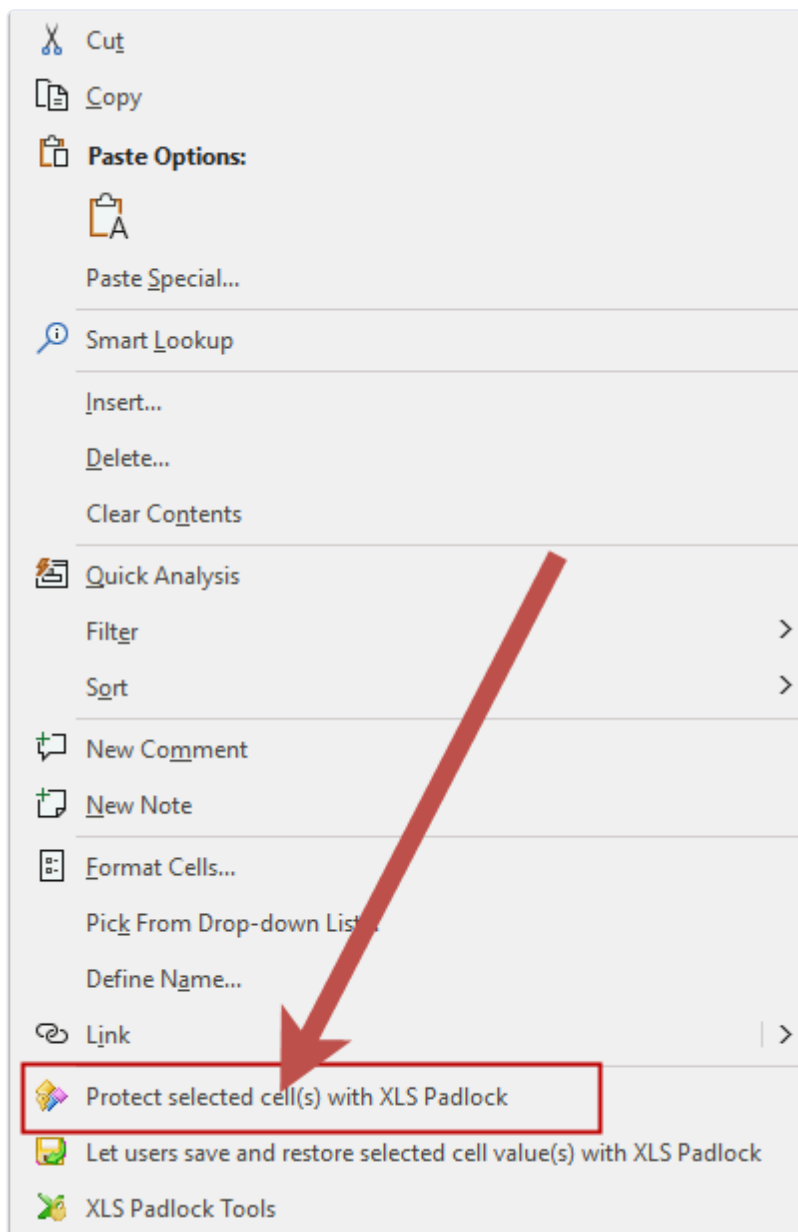
👉 Veja como selecionar as células a serem protegidas

Consulte também:

- [Saiba mais sobre como proteger suas fórmulas de forma eficiente](#)
- [Desativar complementos comuns do Excel](#)
- [Use o VBA Compiler para uma proteção ainda mais forte](#)

Marcar células para proteção de fórmulas

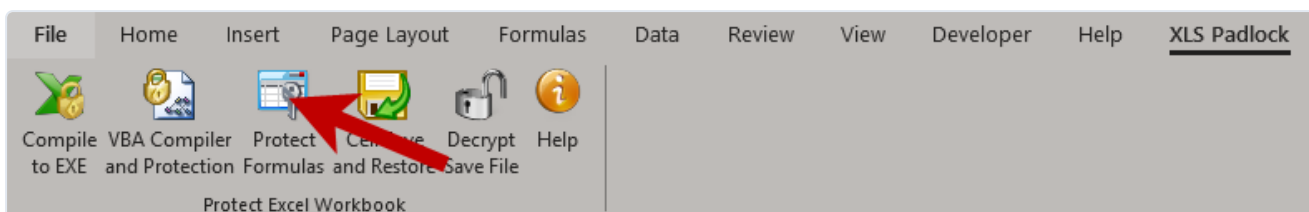
Para proteger uma fórmula, basta **clicar com o botão direito em uma ou mais células** que contenham as fórmulas que você deseja ocultar e selecionar **“Protect selected cell(s) with XLS Padlock”** (proteger as células selecionadas com o XLS Padlock) no menu de contexto.



Em seguida, o XLS Padlock confirmará que as células estão marcadas para proteção.

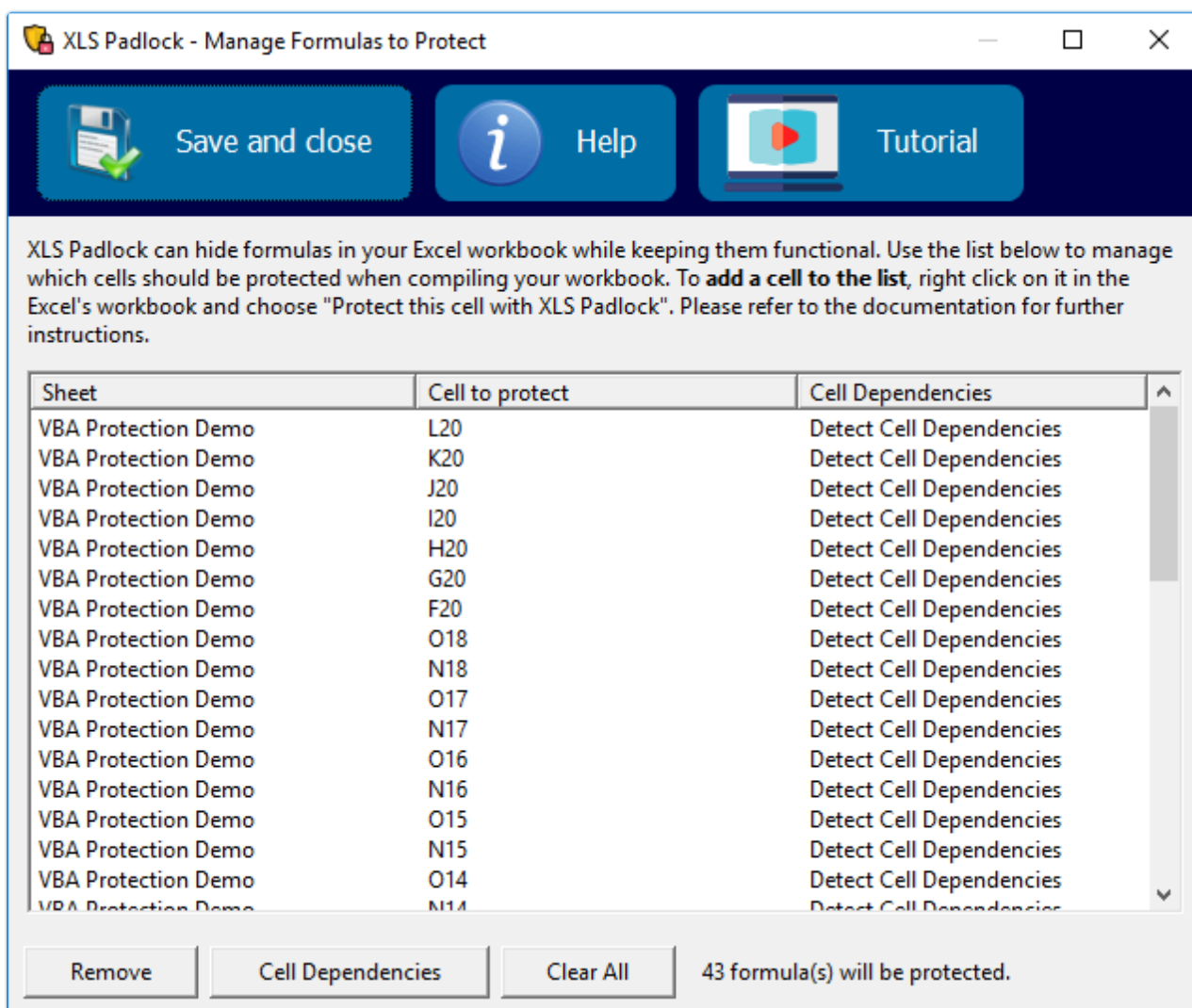
Visão geral das células protegidas

Para ver uma visão geral de todas as células protegidas, clique em **“Protect Formulas”** (proteger fórmulas) na guia ou no menu do XLS Padlock:



Isso abre uma lista de todas as células configuradas para proteção. Nesta janela, você pode ajustar o comportamento da proteção, remover a proteção de células específicas ou limpar a lista inteira.

Quando você compila sua planilha, o XLS Padlock substitui todas as fórmulas listadas por chamadas de funções genéricas como `PLEvalForm(N)` e `PLEvalFormD(N, ...)`. **Suas células continuam funcionais, mas os usuários finais não conseguem descobrir as fórmulas subjacentes.** As fórmulas originais deixam de existir na planilha compilada; elas são gerenciadas pelo próprio EXE.



Dependências de células

O botão **Cell Dependencies** (dependências de células) permite controlar como a proteção é aplicada. Há duas opções disponíveis: “Detect Cell Dependencies” e “No”.

Por padrão, o XLS Padlock detecta todas as referências de células e os nomes de intervalo nas suas fórmulas (dependências de células) e gera uma função anônima que contém essas referências. Isso permite que o Excel recalcule corretamente as células protegidas. Por exemplo, se a fórmula a proteger for `=A3^2`, o XLS Padlock gerará uma função como: `PLEvalFormD(1, COUNT(A3))`.

Se o XLS Padlock não conseguir proteger uma célula, você pode escolher “**No**” na opção “Cell Dependencies” dela. Nesse caso, será usada uma função genérica simples `PLEvalForm(N)`.

Limitações da proteção

Algumas fórmulas complexas não são compatíveis com a proteção e vão falhar, exibindo um `#ERROR!` ou `#VALUE!` na célula. Teste suas fórmulas minuciosamente antes de distribuir sua planilha protegida. Preste atenção especial às seguintes limitações:

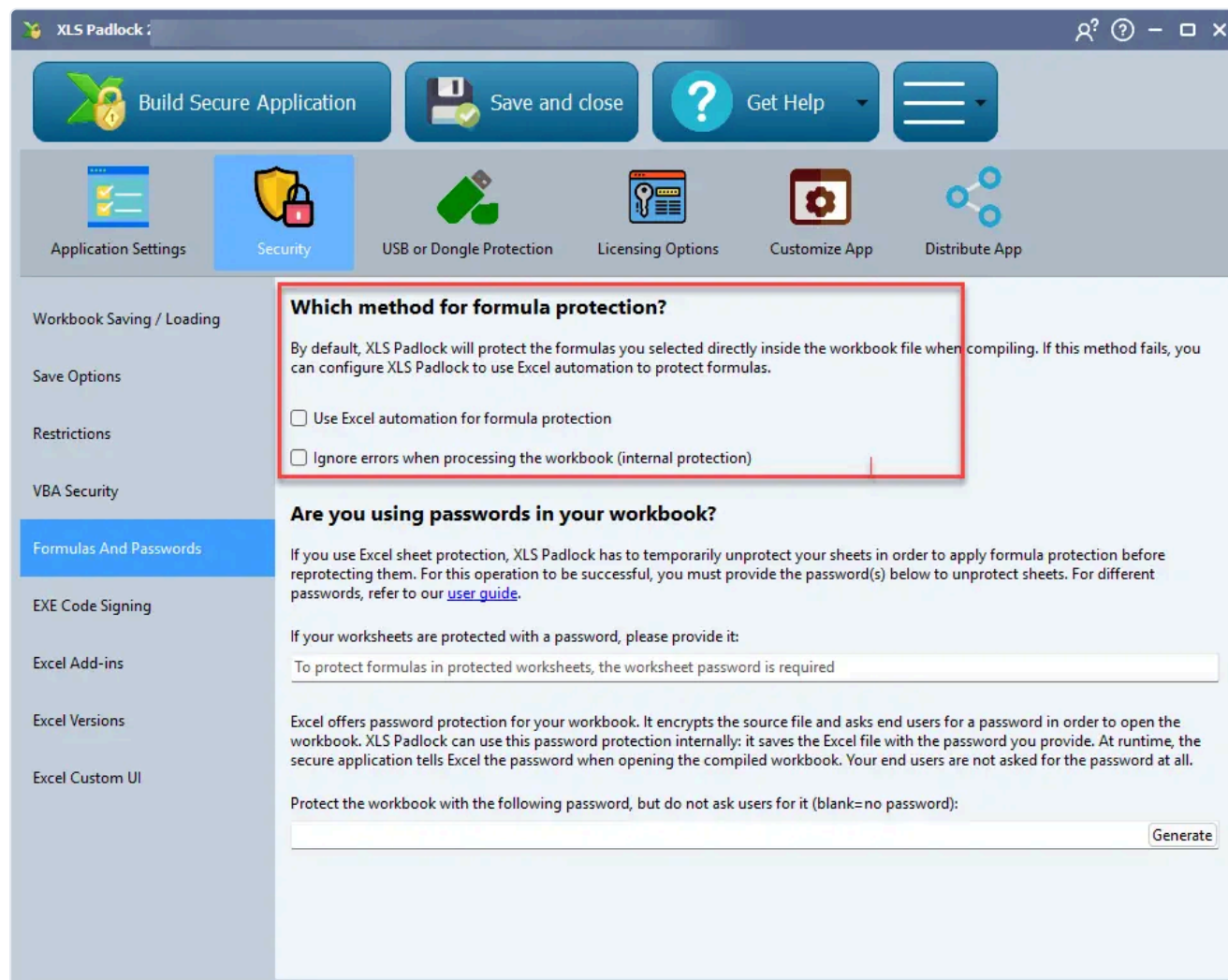
- O comprimento da fórmula deve ser inferior a 256 caracteres.
- A função `INDIRECT` não é compatível.
- As fórmulas não devem conter nenhuma função definida pelo usuário do VBA, apenas funções normais do Excel.
- A proteção de fórmulas deve ser usada com moderação. Proteja apenas as fórmulas mais importantes das quais sua planilha depende. Não se recomenda proteger vários milhares de fórmulas, pois isso aumentará o tamanho do seu arquivo EXE e pode deixar o Excel mais lento em computadores antigos.
- **A validação de dados de células e as regras de formatação condicional não são compatíveis.**

👉 Veja também

- [Combinar a proteção de planilhas do Excel e a proteção do XLS Padlock](#)

Método de proteção de fórmulas

Quando você ativa a proteção de fórmulas, o XLS Padlock pode usar um de dois métodos:



1. **Direct Modification (Recommended):** esse método modifica as fórmulas diretamente dentro do arquivo de origem da planilha. É o método padrão e funciona de forma confiável na maioria dos casos.
2. **Excel Automation:** esse método usa a automação para controlar o Excel e aplicar a proteção.

Se o método recomendado falhar, você pode forçar o XLS Padlock a usar o método de automação ativando "**Use Excel automation for formula protection**". Esse método também é usado automaticamente para planilhas binárias (XLSB) e arquivos muito grandes.

Como último recurso, você pode ativar "**Ignorar erros ao processar a planilha**", mas certifique-se de testar minuciosamente seu aplicativo compilado se ativar essa opção.

Desativar a proteção de fórmulas

Como a proteção de fórmulas do XLS Padlock é incompatível com o Microsoft Excel 2000, você deve ativar a opção **Disable formula and VBA protection** na página de Segurança (Security) se precisar criar aplicativos compatíveis com essa versão.

O VBA Compiler também é desativado

Ativar essa opção também desativará o compilador de código VBA.

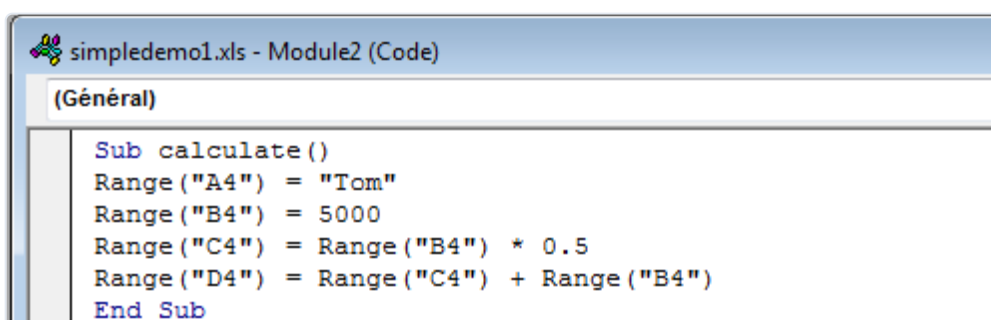
Sobre o VBA Compiler integrado

O XLS Padlock conta com um **compilador VBA integrado** que transforma seus scripts Basic em bytecode seguro, tornando-os inacessíveis aos usuários finais. Compilar partes das suas macros VBA remove o código-fonte original, tornando impossível copiá-lo.

Tutorial em vídeo

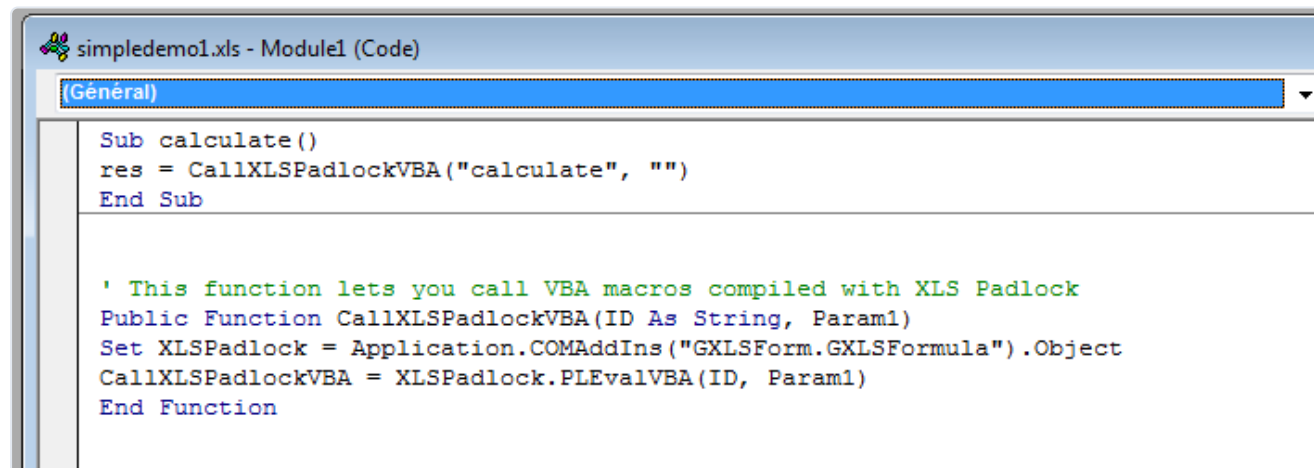
Para ver um tutorial sobre como compilar uma função VBA, você pode [assistir a este vídeo](#).

Por exemplo, 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
```

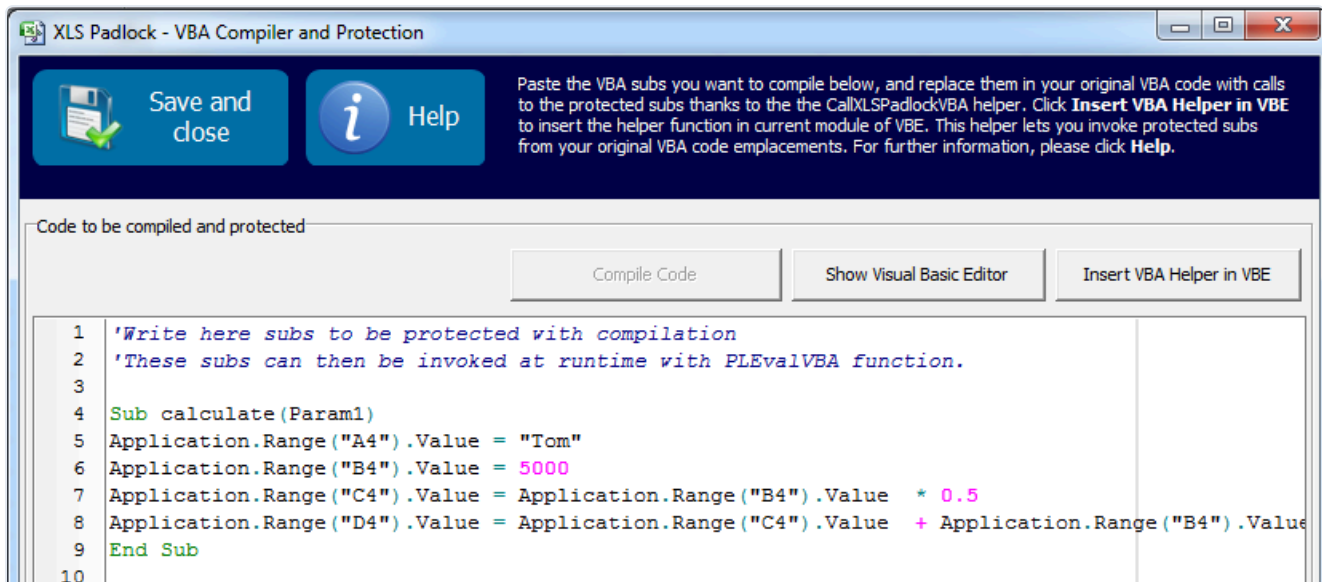
Uma vez protegido, o sub `calculate()` original é substituído por uma chamada a uma função interna do XLS Padlock que executa o bytecode compilado:



```
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
```

O código original é movido para o XLS Padlock VBA Editor e compilado a partir dele:



Observe que algumas modificações podem ser necessárias, já que o compilador exige o uso explícito do objeto `Application` para [acessar os objetos do Excel](#).

O compilador não é um simples ofuscador; ele transforma completamente o seu código VBA em binário e o armazena com segurança dentro do aplicativo. Quando combinado com a [proteção por senha do seu projeto VBA](#), isso torna inúteis as ferramentas de quebra de senha, pois elas não conseguem acessar o arquivo XLS original.

Limitações do compilador

- Pode ser necessário modificar o seu código VBA original para que ele compile, pois o compilador VBA do XLS Padlock não é tão avançado quanto o interpretador VB da Microsoft.
- Em geral, macros inteiras não podem ser compiladas, apenas partes delas.
- Alguns objetos e propriedades padrão não estão acessíveis.

👉 Veja também: [\[Aprenda a escrever e compilar código VBA seguro\]\(#chapter-writing-and-compiling-secure-vba-code\)](#)

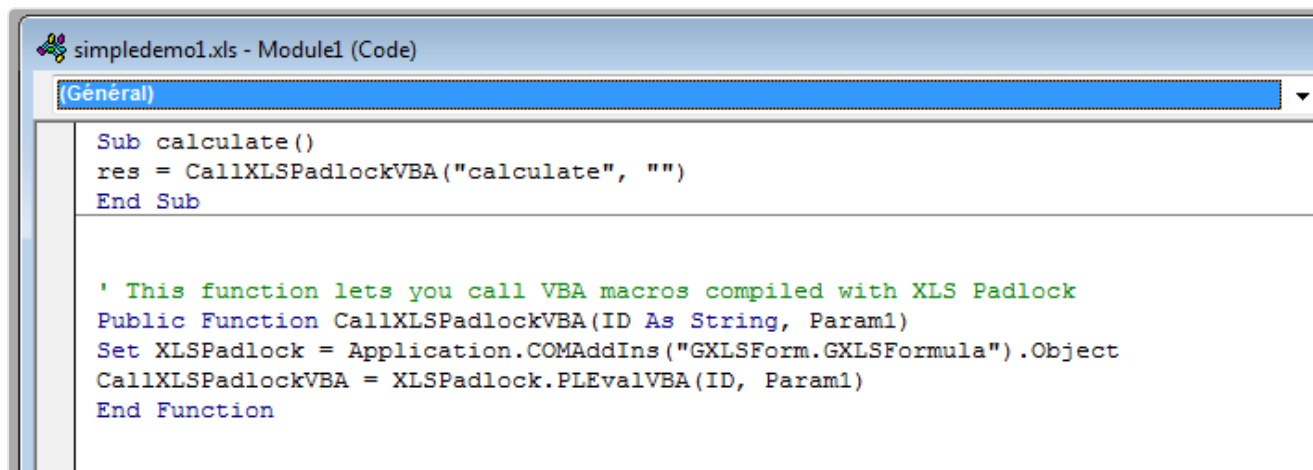
Proteção real do código VBA

Embora algumas ferramentas ofereçam ofuscação de VBA, elas apenas tornam o seu código difícil de ler. Elas não conseguem ocultar a lógica central nem impedir que ela seja copiada.

👉 O XLS Padlock vai além ao permitir que você **remova completamente o código VBA da sua planilha, mantendo-a funcional**. Isso é possível graças ao VBA compiler integrado.

○ **VBA compiler do XLS Padlock converte o seu código VBA em bytecode que só pode ser executado dentro do aplicativo seguro**. Se alguém conseguir [acessar o arquivo da planilha](#), não encontrará o código original, porque ele simplesmente não está lá. Em vez disso, o **bytecode compilado é armazenado de forma segura dentro do EXE**.

Por exemplo, a macro `calculate()` abaixo continha originalmente código VBA proprietário. Após a compilação, a macro na planilha contém apenas uma chamada ao código compilado. A lógica original desapareceu.



```
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
```

Isso oferece uma **proteção robusta**, pois o código VBA original não existe mais na planilha. Ele foi substituído por bytecode que não pode ser copiado nem utilizado em outra planilha do Excel.

A contrapartida é que você deve **transferir manualmente para o editor VBA do XLS Padlock o código VBA que deseja proteger**. A melhor prática consiste em identificar as partes mais críticas do seu código, aquelas que tornariam a sua planilha não funcional se faltassem, e compilar apenas essas seções.

👉 Veja também: [Saiba mais sobre como trabalhar com o nosso VBA compiler](#)

Referência de sintaxe

Este documento fornece uma referência da sintaxe VBA suportada pelo [compilador VBA do XLS Padlock](#).

Estrutura do script

A estrutura do script é formada por declarações de funções e subs.

```
SUB DoSomething
    CallSomething
END SUB

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

As instruções em uma única linha podem ser separadas pelo caractere `:`.

Comentários

É possível inserir comentários dentro de um script. Você pode usar o caractere `'` ou `REM`. O comentário se estende até o final da linha.

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

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

Identificadores

Os nomes de identificadores (variáveis, funções, procedimentos, etc.) devem começar por um caractere (a-z, A-Z) ou um sublinhado `_`. Eles podem ser seguidos por caracteres alfanuméricos ou sublinhados. Os nomes não podem conter nenhum outro caractere nem espaços.

Variáveis

Não é necessário declarar variáveis em um script, mas, se você quiser, pode declarar uma variável por meio da diretiva `DIM`.

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

Você também pode declarar variáveis globais como `PRIVATE` ou `PUBLIC` .

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

As variáveis declaradas com `DIM` são públicas por padrão. As variáveis privadas não são acessíveis a partir de outros scripts. As variáveis também podem ser inicializadas no momento da declaração:

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

Instruções de atribuição

As instruções de atribuição usam o operador `=` para atribuir um valor ou o resultado de uma expressão a uma variável ou a uma propriedade de objeto.

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

Matrizes

O compilador oferece suporte básico a construtores de matrizes e a matrizes do tipo variante. Para construir uma matriz, use os caracteres `[` e `]` . Você pode construir uma matriz de vários índices aninhando construtores de matrizes.

As matrizes no compilador têm 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"
```

Matrizes dinâmicas:

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

Índices

As cadeias de caracteres, as matrizes e as propriedades de matriz podem ser indexadas por meio dos caracteres `[` e `]`. Por exemplo, se `Str` for uma variável de cadeia, a expressão `Str[3]` retorna o terceiro caractere da cadeia.

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

Palavras-chave e operadores

A sintaxe Basic suporta:

- **Declarações:** `SUB...END SUB`, `FUNCTION...END FUNCTION`
- **Diretivas:** `BYREF`, `DIM`
- **Condicionais:** `IF...THEN...ELSE...ELSEIF...END IF`, `SELECT CASE...END SELECT`
- **Loops:** `FOR...TO...STEP...NEXT`, `DO...WHILE...LOOP`, `DO...LOOP...WHILE`, `DO...UNTIL...LOOP`, `DO...LOOP...UNTIL`
- **Operadores:** `^`, `*`, `/`, `AND`, `+`, `-`, `OR`, `<`, `>`, `>=`, `<=`, `=`, `>`, `<`, `DIV`, `MOD`, `XOR`, `SHL`, `SHR`
- **Tratamento de erros:** `TRY...EXCEPT`, `TRY...FINALLY`
- **Outros:** `EXIT`, construtores de matrizes `[1, 2, 3]`, acesso a objetos `ObjectName.Property`

Instruções If

Existem duas formas: `IF...THEN...END IF` e `IF...THEN...ELSE...END IF`. Se a expressão for verdadeira, as instruções do `THEN` são executadas. Se for falsa, as instruções do `ELSE` são executadas (caso existam).

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

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

Se a instrução estiver em uma única linha, você não precisa de `END IF` :

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

Instruções `Select Case`

Se a `selectorExpression` corresponder a uma das expressões `caseexpr` , as instruções respectivas serão executadas. Caso contrário, a instrução `CASE ELSE` será executada.

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

Instruções `For`

A instrução `FOR` repete a execução de instruções até que um contador atinja um valor final.

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

A parte `STEP` é opcional; se for omitida, o valor do passo é 1.

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

Instruções While

Uma instrução `WHILE` repete as instruções enquanto uma condição de controle for verdadeira. A condição é avaliada antes da execução das instruções.

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

Instruções Do ... Loop

As instruções são executadas `WHILE` (enquanto) uma expressão for verdadeira, ou `UNTIL` (até que) uma expressão seja verdadeira. A condição pode ser testada antes ou depois da iteração.

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

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

Declarações de funções e subs

As declarações são semelhantes às do Basic padrão. Para retornar um valor de uma função, atribua-o à variável implícita com o mesmo nome da função, ou use a instrução `Return`. Os parâmetros podem ser passados por referência por meio da diretiva `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
```

Os subs e as funções são públicos por padrão, mas podem ser declarados como `PRIVATE` .

Livro de receitas da API VBA

Este tópico oferece uma coleção completa de trechos de código VBA ("receitas") para o controle avançado da sua aplicação protegida em tempo de execução. Com a API VBA do XLS Padlock, você pode recuperar informações do sistema, gerenciar o salvamento e o carregamento, personalizar a interface do usuário e interagir com recursos online.

Os dois objetos principais da API são: `Application.COMAddIns("GXLSForm.GXLSFormula").Object` para a maior parte das informações em tempo de execução.,

`Application.COMAddIns("GXLS.GXLSPLock").Object` para as operações relacionadas ao salvamento.

Cada receita a seguir fornece uma função ou sub-rotina VBA pronta para uso.

Informações da aplicação e do sistema

Estas receitas ajudam você a recuperar informações sobre o ambiente e o estado da aplicação.

Verificar se a planilha está protegida

Utilize isto para confirmar que o seu código está sendo executado dentro de uma aplicação protegida criada pelo 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
```

Obter o nome do arquivo EXE da aplicação

Recupera o nome do arquivo `.exe` em execução.

```
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
```

Obter a versão do arquivo EXE e do produto

Recupera as cadeias de versão que você definiu na seção "EXE Version Info" do 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")` retorna a versão do produto (Product Version), `ReadVersionInfo("FileVersion")` retorna a versão do arquivo (File Version).

Obter o ID do sistema do usuário

Recupera o ID do sistema exclusivo necessário para as chaves de ativação com bloqueio por hardware. Este é o ID que os seus clientes lhe enviarão.

```
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
```

Obter parâmetros da linha de comando

Recupera os parâmetros da linha de comando passados ao arquivo `.exe` da sua aplicação.

```
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)` retorna o primeiro parâmetro, `ReadParamStr(2)` o segundo, e assim por diante.

Salvar e carregar planilhas

Gerencie os arquivos de salvamento seguros (`.xlsc` ou `.xlsce`) por meio de programação.

Obter o caminho do arquivo de salvamento atual

Recupera o caminho completo para o arquivo de salvamento seguro que está carregado atualmente.

```
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
```

Obter o caminho da pasta de salvamento local

Recupera o caminho para a pasta local onde o XLS Padlock armazena as configurações da sua aplicação e os arquivos de salvamento 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
```

Executar código VBA após salvar

O XLS Padlock pode chamar automaticamente a sub-rotina `XLSPadlock_OnAfterSave` depois que um usuário salva o seu trabalho. Isso é útil para registro em log, para exibir uma mensagem de confirmação ou para outras ações posteriores ao salvamento.

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

Abrir um arquivo de salvamento existente com VBA

Abra um arquivo de salvamento seguro por meio de programação. Observe que isso carregará o arquivo em uma **nova instância do 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 um nome de arquivo para a caixa de diálogo de salvamento

Defina o nome de arquivo padrão que aparece na caixa 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.xlsx"
' Call SetSecureWorkbookFilename("D:\My Documents\my save.xlsx")
```

Salvar uma cópia segura sem solicitar confirmação

Salve uma cópia segura da planilha diretamente em um caminho de arquivo especificado sem exibir a caixa 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.xlsx") Then
'     MsgBox "Workbook saved successfully."
' End If
```

Migrar dados de usuário de uma versão anterior

Se você publicar uma nova versão da sua planilha, talvez precise importar dados de um arquivo de salvamento criado com uma versão anterior.

→ [Saiba como migrar dados de usuário com VBA](#)

Interface do usuário

Personalize elementos da interface, como caixas de diálogo.

Ocultar por meio de programação as caixas de diálogo de espera/carregamento

Você pode ocultar a caixa de diálogo "Loading workbook, please wait..." com uma chamada VBA em vez de esperar que ela se feche automaticamente.

```
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
```

Ativação e validação online

Interaja com os recursos online do XLS Padlock.

Verificar se há conexão com a Internet

Testa se há uma conexão ativa com a Internet disponível no computador do usuário.

```
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
```

Obter o token de ativação online

Recupera um hash do token de ativação retornado pelo servidor web após uma ativação online bem-sucedida.

```
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
```

Verificar se a validação online foi bem-sucedida

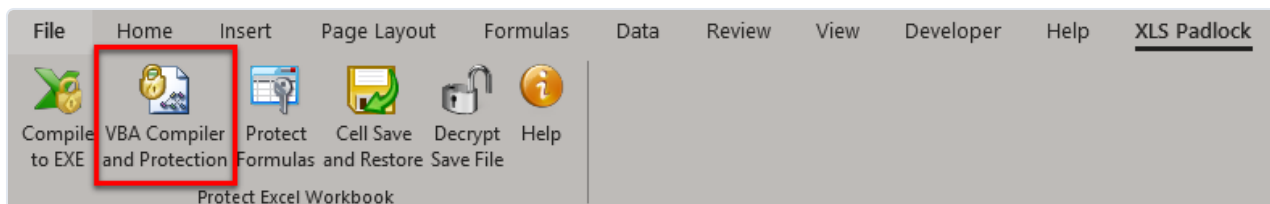
Obtém o resultado da última tentativa de validação online.

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

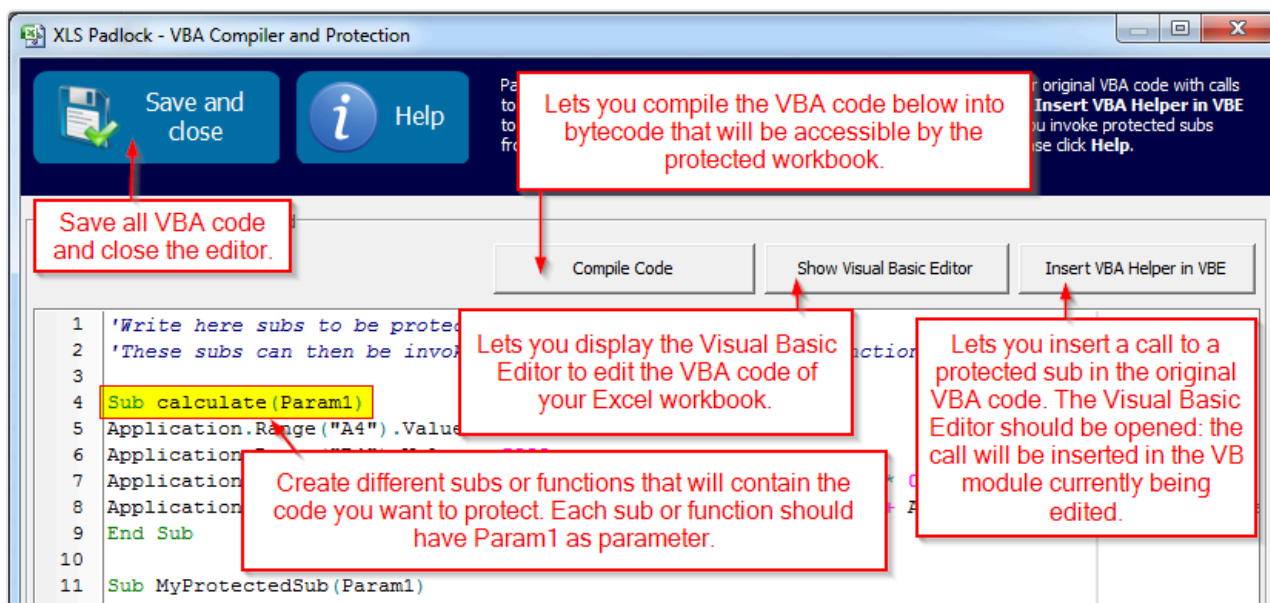
Escrever e compilar VBA seguro

Você pode **proteger as partes sensíveis do seu código VBA compilando-as em bytecode com o XLS Padlock**. Recomendamos combinar o bytecode compilado com seu código VBA existente para obter uma abordagem equilibrada entre segurança e desenvolvimento.

1. No Excel, clique em "VBA Compiler and Protection" para abrir o editor e compilador VBA do XLS Padlock:



O editor VBA do XLS Padlock será aberto:



2. Na área de texto, digite o código VBA que deseja compilar. Você pode agrupar seu código em `subs` e `functions`. Para simplificar esse processo, você pode abrir o Editor do Visual Basic clicando em "Show Visual Basic Editor" e copiar e colar seu código.

Por padrão, todos os subs e funções têm um parâmetro chamado `Param1`. No entanto, você pode usar mais parâmetros.

3. Quando terminar, clique em "**Compile Code**". O código VBA é instantaneamente compilado em bytecode, que o XLS Padlock armazenará até que você compile o arquivo EXE do seu aplicativo.

DICA

Se um erro for detectado, ele será exibido e o código correspondente ficará sublinhado em vermelho.

1. Clique em "Save and close" para fechar o editor.

👉 Após compilar seu código, você precisará **modificar suas macros VBA originais para que chamem a versão compilada**. Saiba como fazer isso no guia [Invocar código VBA compilado em tempo de execução](#).

Invocar o VBA compilado em tempo de execução

Para executar seu [código VBA compilado](#), você deve chamá-lo a partir do código VBA normal da sua planilha usando uma função auxiliar.

Primeiro, abra o Visual Basic Editor e cole a seguinte função auxiliar em um módulo da sua planilha:

```
' 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
```

Inserção automática

Esta função pode ser inserida automaticamente no seu módulo VB ativo clicando no botão "Insert VBA Helper in VBE" (inserir auxiliar VBA no VBE) do editor VBA do XLS Padlock.

A função auxiliar `CallXLSPadlockVBA` recebe dois parâmetros:

- **ID:** O nome do `Sub` ou `Function` compilado que você deseja invocar.
- **Param1:** Um parâmetro opcional que você deseja passar para o código compilado.

Exemplo

Imagine que seu módulo VBA original contém o seguinte código:

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

Depois de mover este código para o VBA Compiler, você substituiria o sub original por uma chamada à função auxiliar:

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

Aqui, fornecemos o nome do sub protegido, `Calculate`, e passamos uma string vazia como segundo parâmetro porque ele não é utilizado neste caso. A variável `res` conterá o resultado da chamada, o que é útil principalmente quando o código invocado é uma `Function` que retorna um valor.

Execução em tempo de design

Um dos recursos mais poderosos do XLS Padlock é que essa configuração funciona até mesmo em tempo de design. Se você executar o sub `Calculate()` a partir do VBE antes de compilar a planilha, o XLS Padlock executará o código protegido, permitindo testar sua lógica sem precisar compilar a aplicação primeiro.

👉 Veja também: Aprenda a [passar mais de um parâmetro para o seu código compilado](#chapter-passing-more-parameters-to-the-compiled-vba-code).

Acessar objetos do Excel

Ao escrever código para o compilador VBA, você deve usar explicitamente o objeto `Application` para acessar o modelo de objetos do Excel.

Para uma referência detalhada das propriedades e dos métodos disponíveis, consulte a [documentação do Microsoft Office Dev Center](#).

Exemplos

Referenciar um intervalo (Range)

Para referenciar uma célula ou um intervalo de células na sua planilha, você deve anteceder o objeto `Range` com `Application`.

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

Usar funções de planilha

Para usar as funções de planilha do Excel, você deve usar o objeto `Application.WorksheetFunction`.

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

As funções exigem todos os parâmetros

Ao chamar funções internas do Excel como `InputBox` a partir de código compilado, você **deve especificar todos os parâmetros**, pois os parâmetros padrão não são compatíveis.

Exemplo completo: converter uma macro

Aqui está uma macro VBA padrão:

```
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
```

Aqui está o mesmo código modificado para o compilador VBA do XLS Padlock. Observe a adição do objeto `Application` e os parâmetros 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
```

Seu módulo VBA original seria então modificado para chamar a versão compilada:

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

Passar matrizes

Você pode passar diferentes tipos de variáveis para o código VBA compilado, incluindo arrays estáticos.

Suponha que você tenha a seguinte função no VBA Compiler:

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

No seu módulo VBA normal do Excel, você pode chamar essa função e passar um array.

Aviso

O array deve ser definido como um **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
```

Passar mais parâmetros

Por padrão, a função auxiliar usada para executar o código VBA compilado aceita apenas um parâmetro.

```
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
```

O método `PLEvalVBA` do objeto `XLSPadlock` recebe dois parâmetros: o **ID** do sub/função compilado e um único parâmetro **Param1**.

Para passar mais parâmetros, o objeto `XLSPadlock` fornece métodos adicionais:

- `XLSPadlock.PLEvalVBA2(ID, Param1, Param2)` para dois parâmetros.
- `XLSPadlock.PLEvalVBA3(ID, Param1, Param2, Param3)` para três parâmetros.

Também é possível [passar mais parâmetros usando arrays](#).

Por exemplo, uma função auxiliar para passar dois parâmetros teria 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
```

Tratar erros

Erros em tempo de execução podem ocorrer quando o seu código VBA compilado é executado. Enquanto o VBA padrão utiliza a instrução `On Error` para o tratamento de erros, o VBA compiler do XLS Padlock utiliza um bloco `Try...Except`.

Quando um erro ocorre dentro do bloco `Try` (ou em qualquer procedimento que ele chame), o compilador saltará imediatamente para o bloco `Except` para o tratamento.

Sintaxe

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

DICA

Um erro não tratado será exibido pelo VBA compiler em tempo de execução, [a menos que você desative essa opção](#).

Exemplo 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
```

OLE error 800A03EC

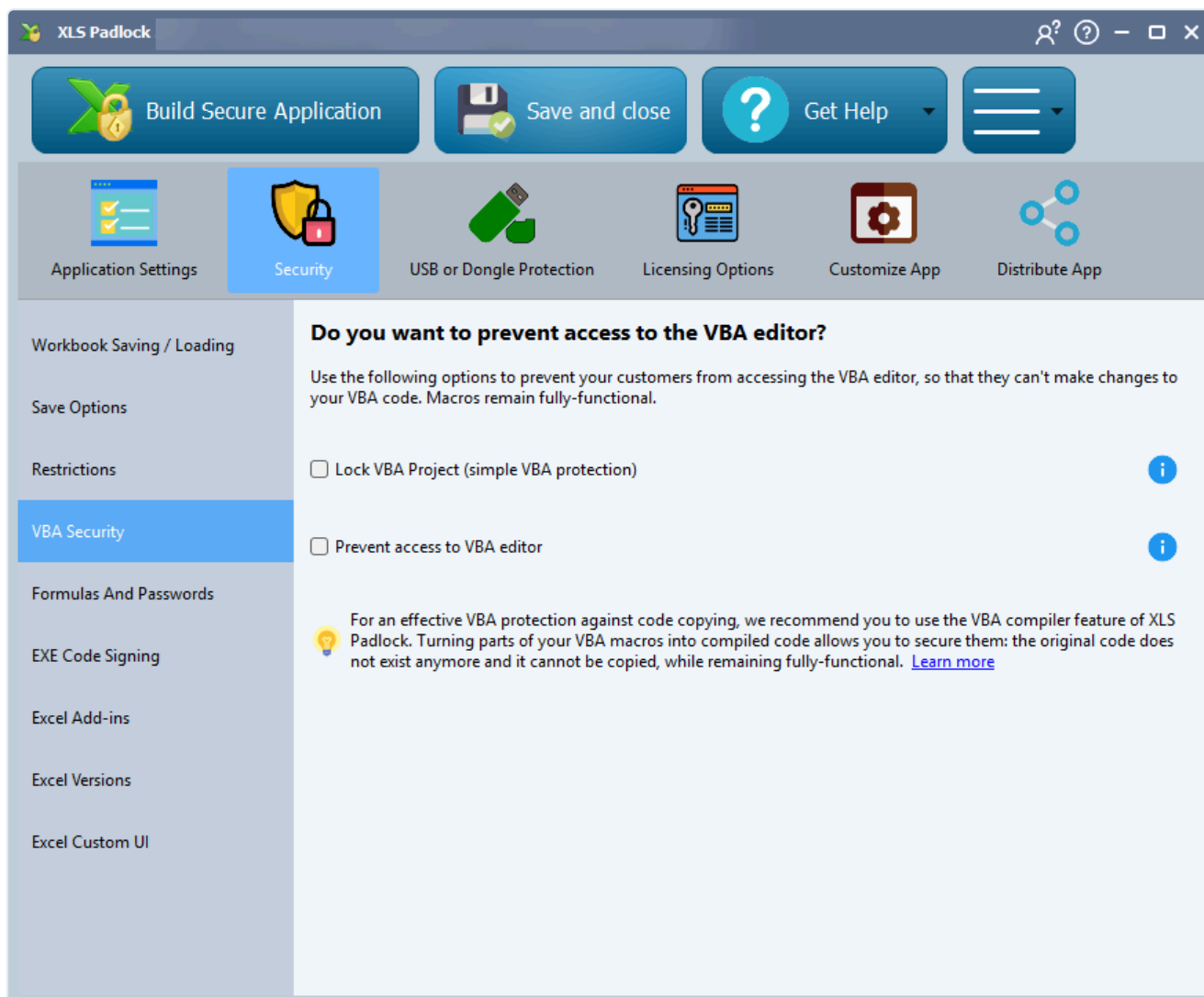
Para resolver esse erro, verifique se a opção **“Trust access to the VBA project object model”** está ativada nas configurações da Central de Confiabilidade (Trust Center) do Excel.

Para obter instruções sobre como localizar essa opção, consulte a página oficial de suporte do Microsoft Office:

[Ativar ou desativar macros em documentos do Office](#)

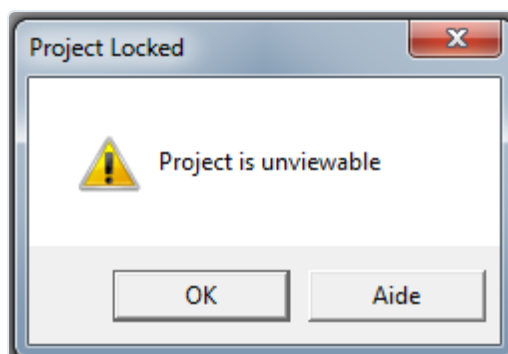
Proteção do código VBA

O seu código VBA é um ativo valioso. O XLS Padlock fornece várias camadas de proteção para protegê-lo contra adulteração ou roubo. A seguir, são apresentadas duas opções de segurança fundamentais, seguidas da nossa solução mais avançada para a máxima proteção.



Opção 1: bloquear o seu projeto VBA

O passo mais simples é [bloquear o seu projeto VBA](#). Isso faz com que o seu código não possa ser visualizado no Visual Basic Editor. Se um usuário tentar acessar o projeto, ele encontrará uma mensagem de erro, o que bloqueia de forma efetiva a inspeção casual do seu código.



Opção 2: impedir o acesso ao editor de VBA

Para uma camada de segurança adicional, você pode [impedir totalmente que o editor de VBA seja aberto](#). O XLS Padlock monitorará continuamente e fechará automaticamente qualquer tentativa de abrir o VBE, garantindo que até mesmo usuários experientes sejam bloqueados.

Proteção definitiva: compilação de VBA

Para o mais alto nível de segurança, o XLS Padlock permite que você **compile o seu código VBA em código binário nativo**. Ao contrário da simples ofuscação, que apenas torna o código difícil de ler, a compilação transforma a sua lógica em um formato não legível por humanos que é quase impossível de submeter a engenharia reversa. O seu código VBA original é removido da planilha e substituído por chamadas ao código compilado e seguro dentro do EXE. Esta é a defesa definitiva para a sua propriedade intelectual.

→ [Saiba mais sobre o nosso poderoso VBA Compiler](#)

Ocultar e bloquear o código VBA

O XLS Padlock oferece um recurso dedicado [Lock VBA Project](#) (Bloquear projeto VBA). No entanto, se você optar por não utilizá-lo, ainda pode recorrer à proteção nativa do Excel. O XLS Padlock torna esse método mais seguro.

Como bloquear o seu projeto VBA no Excel

1. No Excel, abra o Visual Basic Editor (VBE) pressionando `Alt+F11` ou selecionando-o na guia `Developer`.
2. No VBE, vá até o menu **Tools** (Ferramentas) e selecione **VBAProject Properties...** (Propriedades do VBAProject).
3. Na caixa de diálogo, vá até a aba **Protection** (Proteção).
4. Marque a caixa **Lock project for viewing** (Bloquear projeto para visualização) e insira uma senha forte.

Depois que a planilha for salva e reaberta, os usuários não poderão visualizar nem editar o código VBA sem a senha.

Por que isso é mais seguro com o XLS Padlock

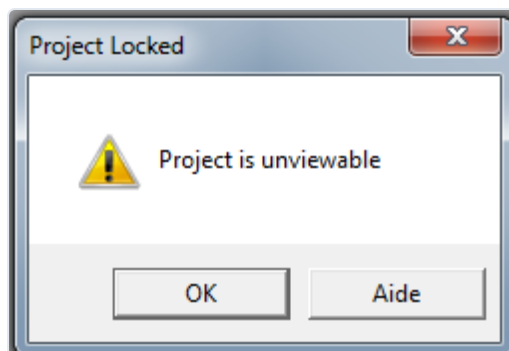
Normalmente, a proteção por senha do VBA do Excel é considerada fraca, porque existem muitas ferramentas capazes de remover a senha. Essas ferramentas funcionam acessando diretamente o arquivo do Excel.

No entanto, quando a sua planilha é compilada em um EXE com o XLS Padlock, o arquivo `.xls` ou `.xlsm` original é criptografado e incorporado dentro da aplicação. As ferramentas de quebra de senha não conseguem acessar esse arquivo protegido, o que as torna inúteis. Essa combinação proporciona uma defesa muito mais robusta para o seu código VBA.

Bloquear o projeto VBA

A opção **Lock VBA Project** oferece uma forma simples de impedir que os usuários finais acessem o seu projeto VBA. Esse recurso marca o projeto VBA como bloqueado, tornando impossível visualizá-lo, acessá-lo ou modificá-lo. Não utiliza proteção por senha.

Se o usuário final tentar acessar um projeto VBA bloqueado, a seguinte mensagem de erro é exibida (Project Locked, Project is unviewable):



Remova as senhas VBA existentes

Se você usar esta opção, recomendamos remover qualquer senha existente do seu projeto VBA. Uma senha se torna redundante quando o projeto está bloqueado. Além disso, usar uma senha longa com esta opção pode causar travamentos no Excel 2007.

Se preferir manter uma senha, considere usar a opção [Prevent access to VBA editor](#chapter-forbid-access-to-the-vba-editor-vbe) (impedir o acesso ao editor VBA).

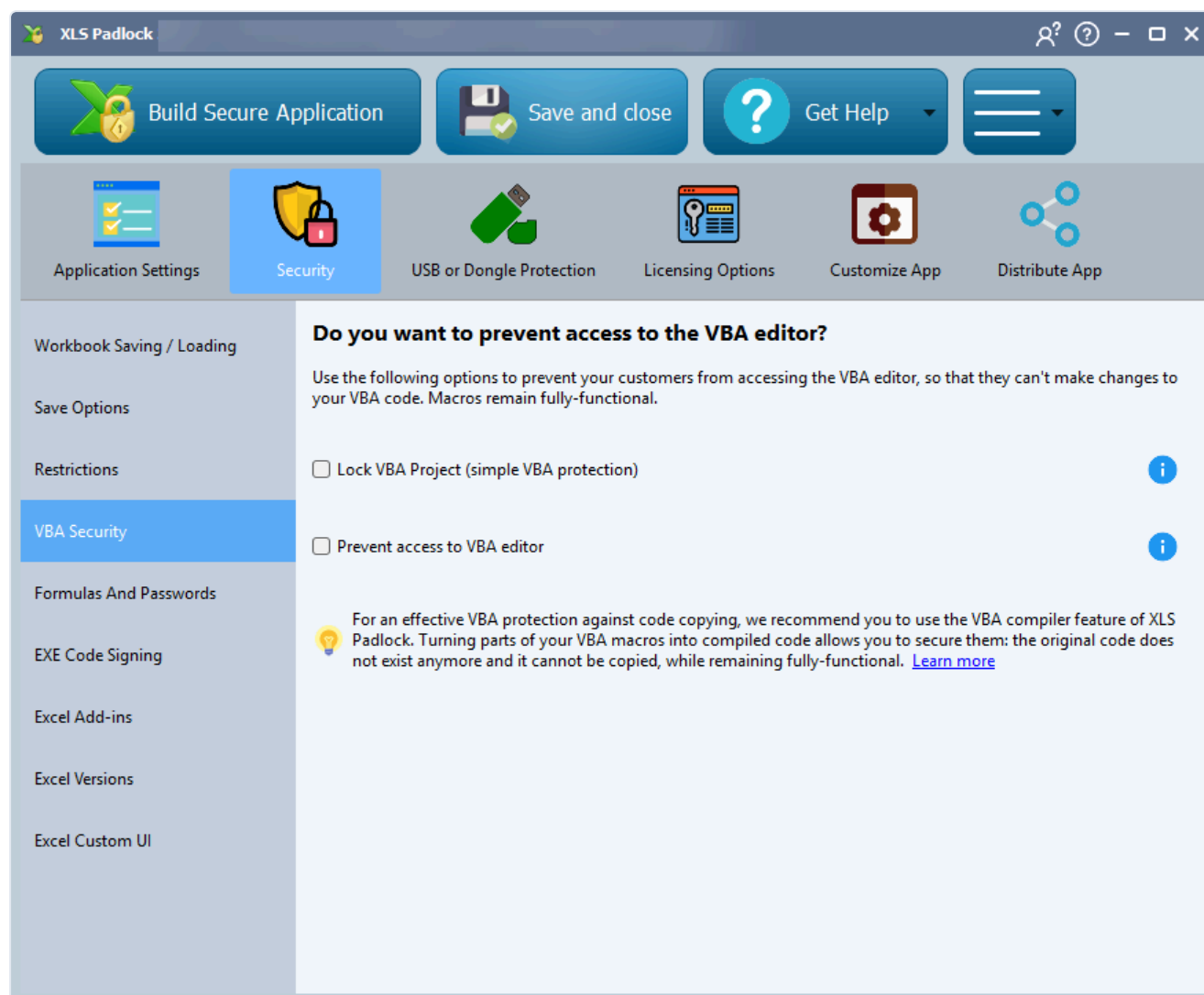
Observe que as macros continuam funcionais.

Esta opção é compatível com o nosso [compilador VBA](#), que oferece um nível de segurança superior para o seu código VBA.

Proibir o acesso ao VBE

O XLS Padlock oferece opções para **impedir que seus clientes acessem o editor VBA (VBE)**, impedindo assim de forma eficaz que eles visualizem ou alterem seu código VBA, ao mesmo tempo que garante que suas macros permaneçam plenamente funcionais.

Essas opções estão disponíveis na aba "Security" do XLS Padlock.



Lock VBA Project (proteção VBA simples)

Essa opção bloqueia seu projeto VBA por meio de programação para dificultar o acesso. Embora ofereça um nível básico de proteção, ela pode ser contornada por algumas ferramentas.

Prevent Access to VBA Editor

Essa é uma medida de segurança mais forte. O XLS Padlock realizará uma verificação periódica e, se um usuário tentar abrir o Visual Basic Editor (VBE), o XLS Padlock fechará imediatamente a janela do VBE. Isso impede que ele sequer tente visualizar o código.

Essa opção é compatível com a opção "Lock VBA Project" e com nosso poderoso [compilador VBA](#).

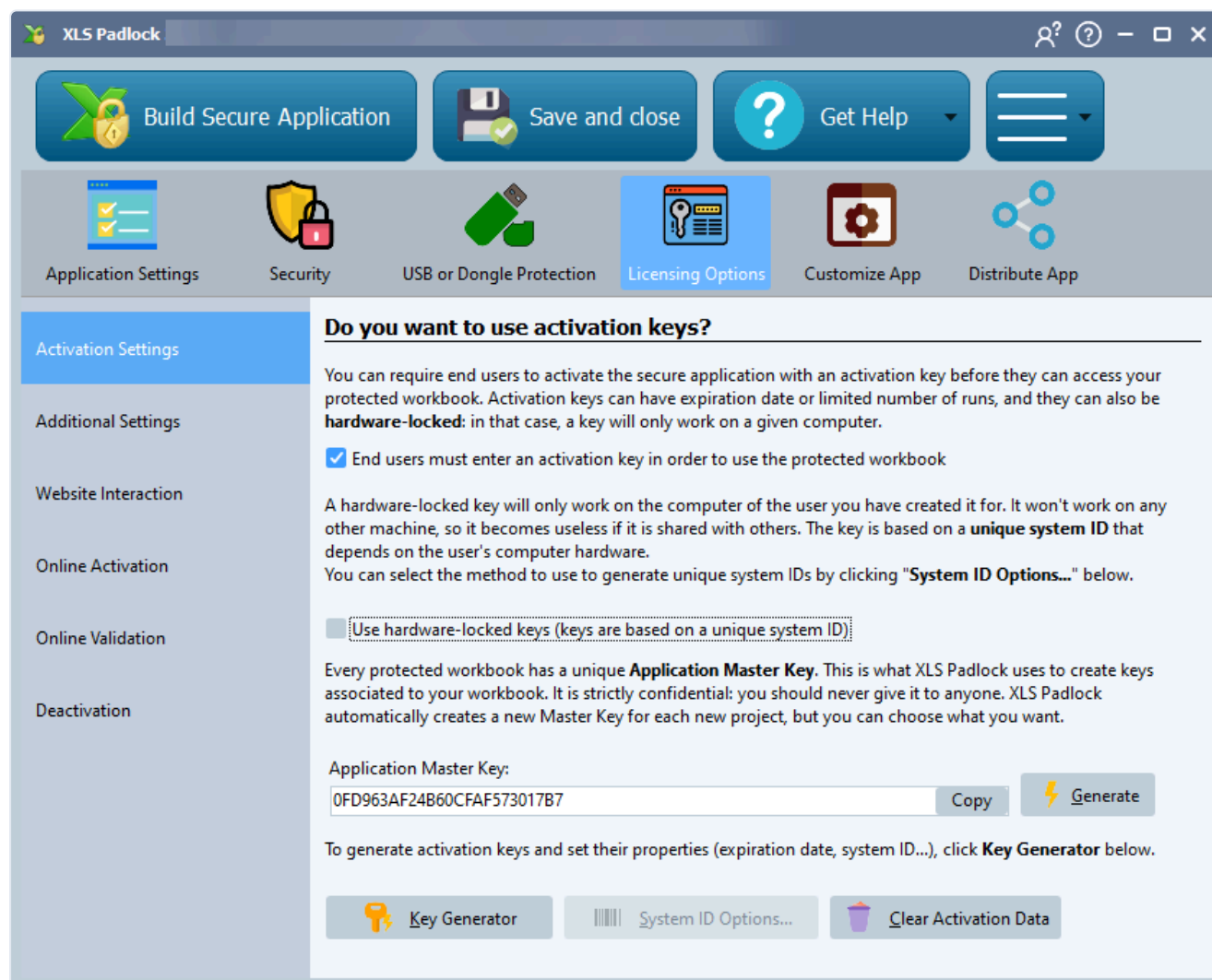
Para a máxima proteção do código VBA

👉 Para uma proteção robusta contra o roubo de código e a engenharia reversa, recomendamos fortemente utilizar o [recurso de compilador VBA](#).

Ao compilar partes das suas macros VBA, o código-fonte original é removido e substituído por código binário. Esse código compilado é extremamente difícil de submeter a engenharia reversa, garantindo que seus algoritmos e sua lógica de negócio fiquem seguros. É a forma mais eficaz de proteger sua propriedade intelectual VBA.

Guia de ativação e licenciamento

Este guia aborda as diversas opções relacionadas a **chaves de ativação, licenciamento, validação online e desativação** para seus aplicativos protegidos.



Conceitos básicos

Habilitação das chaves de ativação

Para exigir que os usuários tenham uma chave para usar seu aplicativo, habilite a opção: **"End users must enter an activation key in order to use the protected workbook"** (os usuários finais devem inserir uma chave de ativação para usar a planilha protegida). Esse é o interruptor principal para ativar os recursos de licenciamento.

Application Master Key

Toda planilha protegida tem uma **Application Master Key** exclusiva. O XLS Padlock usa essa chave para gerar as chaves de ativação associadas à sua planilha. Ela é **estritamente confidencial** e nunca deve ser compartilhada. O XLS Padlock cria automaticamente uma nova Master Key para cada projeto.

Application GUID e Secret Key

- O **Application GUID** é um identificador exclusivo que o aplicativo usa para armazenar suas configurações e gerenciar os arquivos salvos no computador do usuário.
- A **Application Secret Key** é usada para criptografar e proteger os arquivos salvos pelo usuário (`.XLSC` ou `.XLSCE`). Ela garante que os arquivos salvos com seu aplicativo só possam ser abertos pelo seu aplicativo.

IMPORTANTE

Se você alterar o Application GUID ou a Secret Key, os arquivos salvos anteriormente deixarão de ser acessíveis para seus usuários.

Publicar uma atualização que deve invalidar os arquivos salvos antigos dos usuários

Para forçar uma nova versão a ignorar qualquer arquivo salvo com versões anteriores, por exemplo, quando a estrutura da planilha mudou o suficiente para que os arquivos salvos antigos não devam ser recarregados, gere uma **nova Application Secret Key** antes de recompilar, mas **deixe o Application GUID inalterado** para que as chaves de ativação existentes permaneçam válidas. Na primeira execução do usuário, o arquivo salvo antigo será detectado como incompatível e ele será solicitado a carregar a planilha original incorporada em seu lugar. Consulte [Atualizações da planilha](#) para conhecer o fluxo de trabalho completo de atualização.

DICA

O **status de ativação (avaliação / registrado)** pode ser obtido por meio de código VBA.

Permitir que o usuário final altere a chave de ativação

Um usuário pode alterar sua chave de ativação executando o aplicativo com o **parâmetro de linha de comando** `-enterkey` (por exemplo, `MYAPP.EXE -enterkey`). Isso é útil para substituir uma chave que está prestes a expirar.

Limpar os dados de ativação

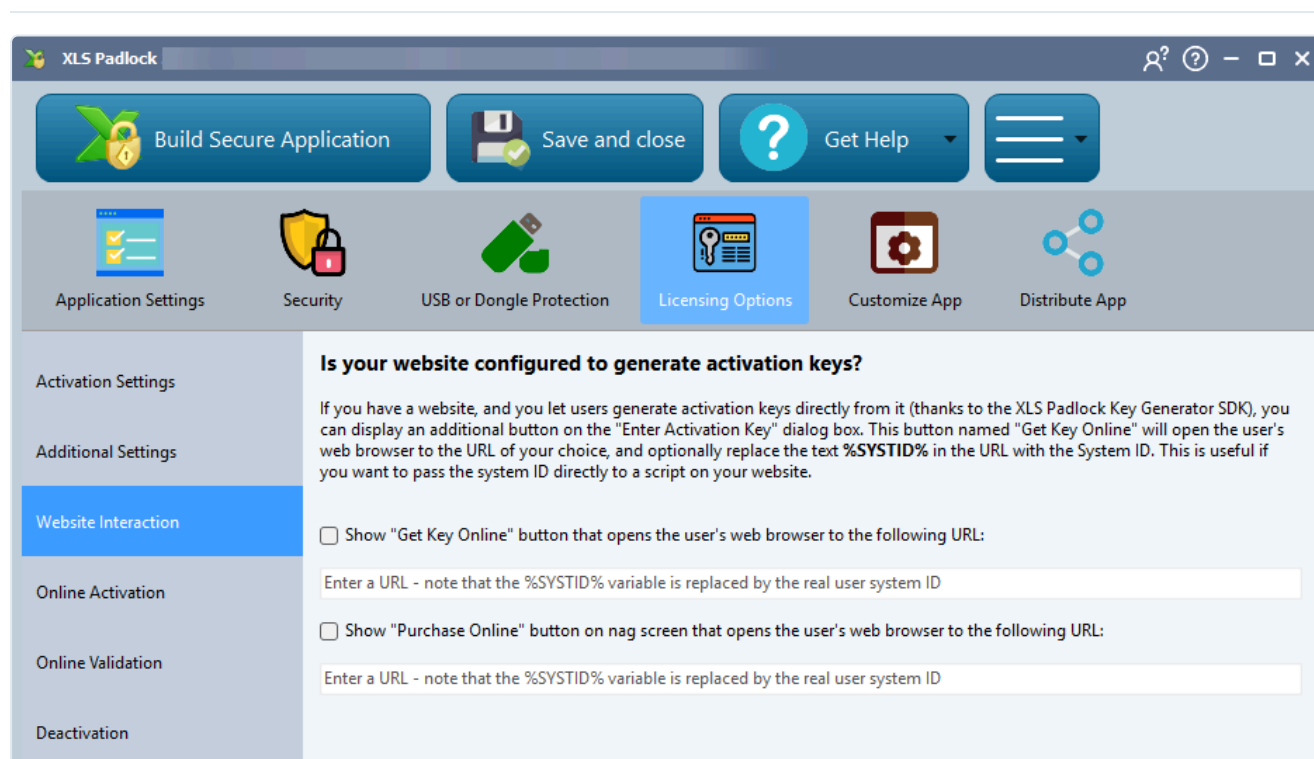
Para fins de teste em sua própria máquina, o botão "**Clear Activation Data**" (limpar dados de ativação) do Key Generator redefine todas as informações de ativação armazenadas localmente.

Chaves com bloqueio por hardware

Uma chave com bloqueio por hardware **só funcionará no computador específico** para o qual foi gerada, tornando-a inútil se compartilhada. A chave se baseia em um **System ID exclusivo** derivado do hardware do computador do usuário.

- **Opções de System ID:** você pode selecionar quais componentes de hardware (CPU, endereço MAC, etc.) são usados para gerar o System ID. Seja consistente com essas opções após implantar seu aplicativo.
- **Obtenção do System ID:** quando um usuário executa o aplicativo, será exibido o System ID dele, que ele deverá enviar a você. Em seguida, você usa esse ID no Key Generator para criar a chave dele.
- **Automação online:** para evitar a troca manual de System IDs, você pode usar os recursos de [ativação online](#).

Interação com o site



Botão Get Key Online

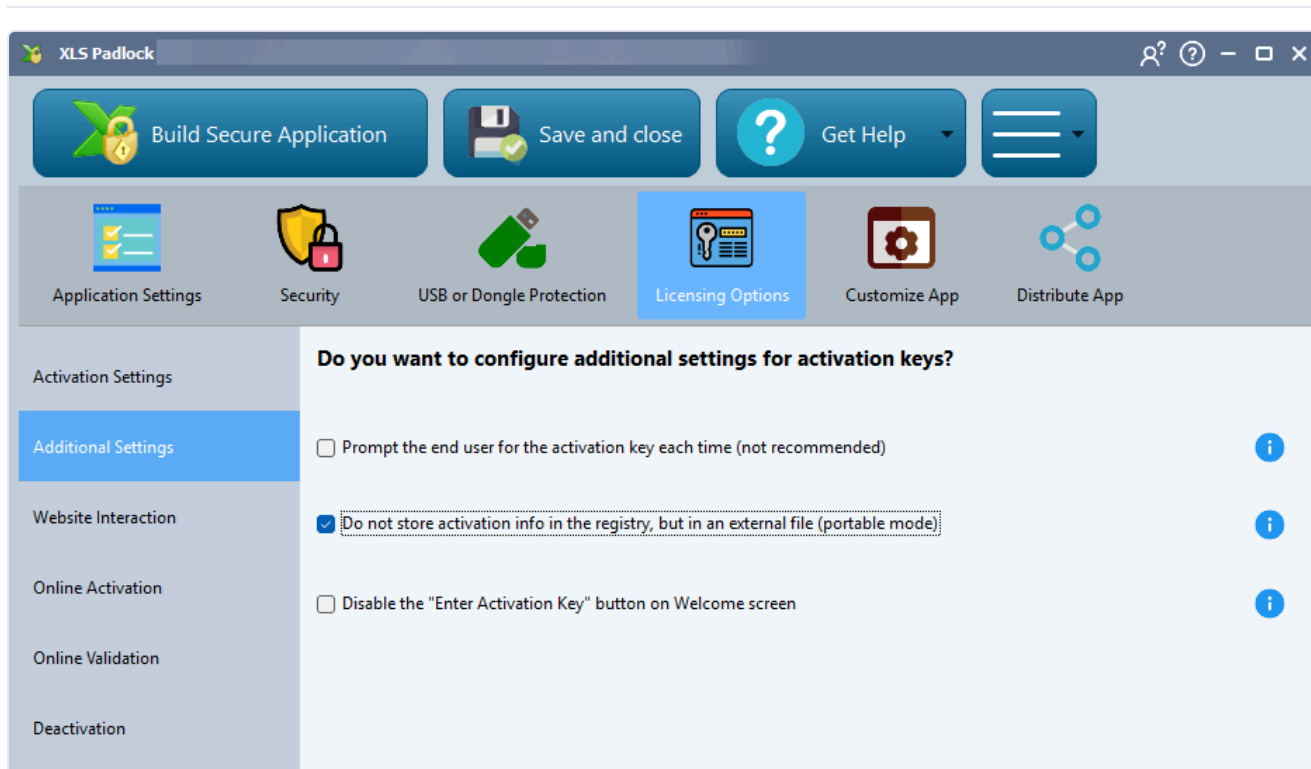
Você pode exibir um botão "Get Key Online" (obter chave online) na janela de ativação. Esse botão abre o navegador da web do usuário em uma URL que você especificar. Você pode usar o marcador de posição `%SYSTID%` na URL para passar automaticamente o System ID do usuário ao script de geração de chaves do seu site.

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

Mostrar o botão "Purchase Online"

Para versões de avaliação com uma tela de lembrete, você pode adicionar um botão "Purchase Online" (comprar online) que direciona o usuário à sua loja ou página de compra.

Configurações adicionais



Modo portátil

Habilite a opção **"Do not store activation info in the registry, but in an external file (portable mode)"** (não armazenar informações de ativação no registro, mas em um arquivo externo, modo portátil). Isso cria um arquivo `.lic` oculto na mesma pasta do seu EXE, permitindo que o aplicativo seja executado a partir de uma unidade USB sem gravar no registro do Windows.

IMPORTANTE

Esse modo não é recomendado para versões de avaliação, pois excluir o arquivo `.lic` redefinirá o período de avaliação.

Solicitar a chave de ativação ao usuário final a cada vez

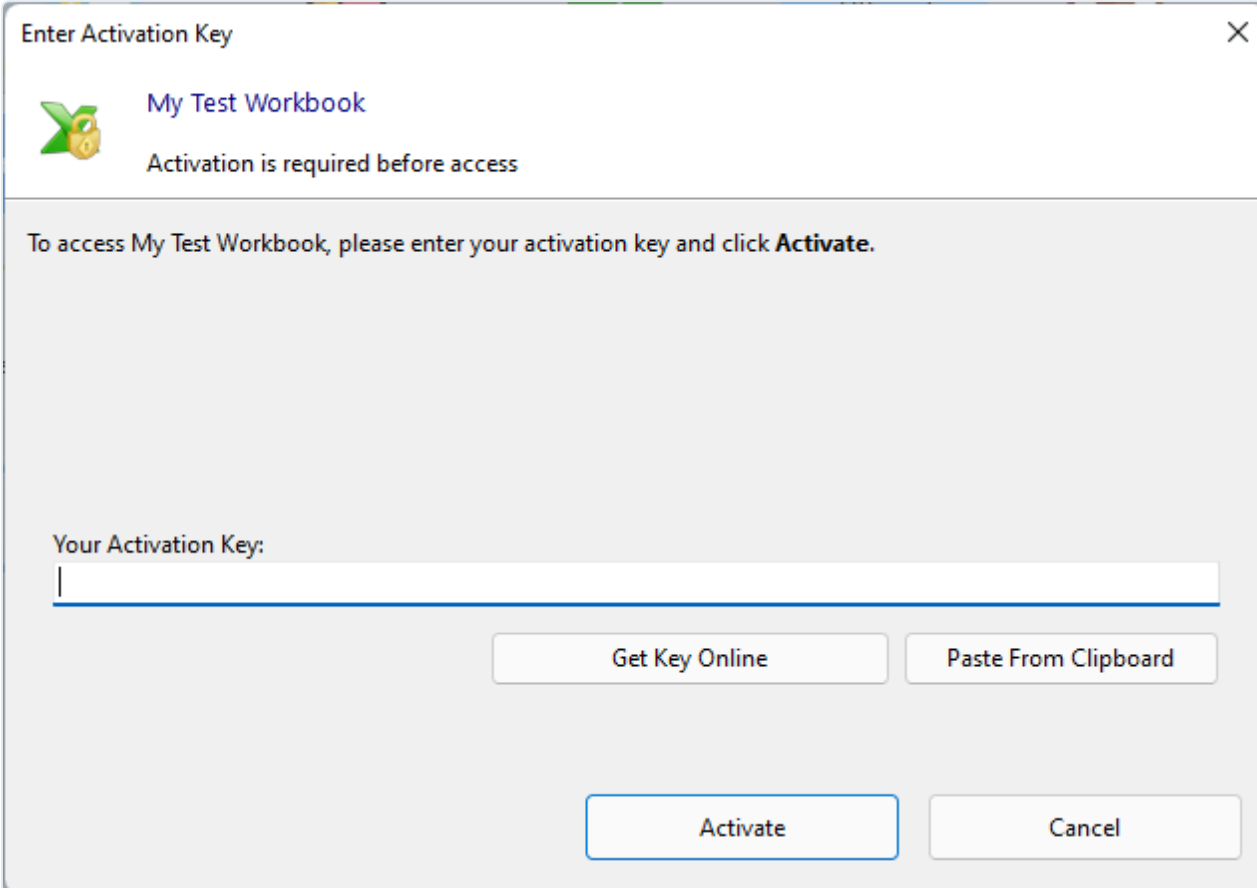
Ao habilitar "Prompt the end user for the activation key each time" (solicitar a chave de ativação ao usuário final a cada vez), seus usuários terão que inserir a chave de ativação sempre que abrirem seu aplicativo. Não é recomendado, pois é cansativo para o seu cliente e você não pode definir recursos de expiração para a chave.

Desabilitar o botão "Enter Activation Key" na tela de boas-vindas

Isso removerá a possibilidade de seus usuários finais inserirem uma chave de ativação ao abrirem seu aplicativo de planilha do Excel. No entanto, não é recomendado usar essa opção.

Chaves de ativação

As chaves de ativação são um recurso essencial do XLS Padlock, oferecendo um **potencioso sistema de licenciamento para suas planilhas do Excel**. Esse sistema permite que você exija que os usuários finais ativem o aplicativo com uma chave antes de poderem acessar sua planilha protegida:



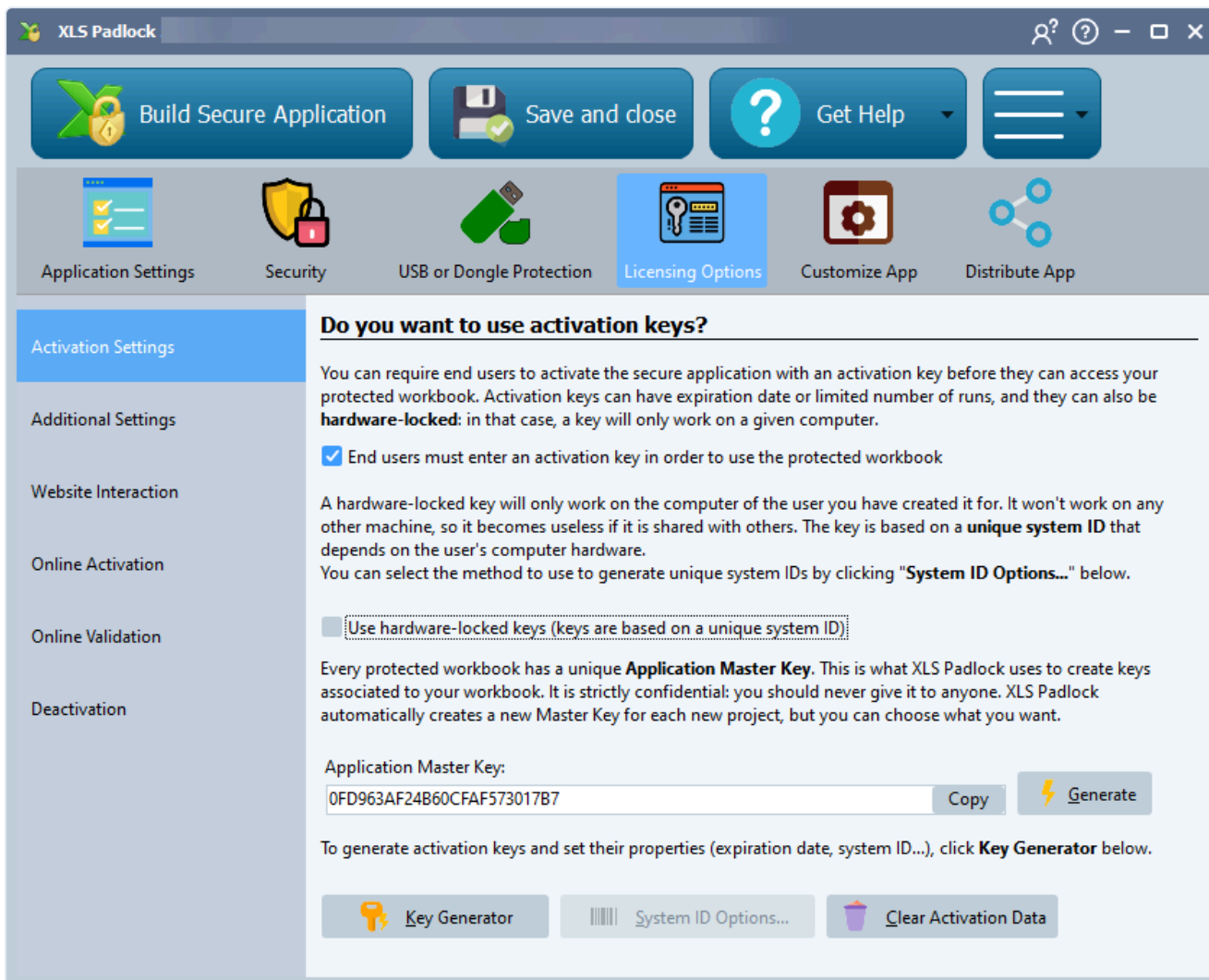
The screenshot shows a dialog box titled "Enter Activation Key" with a close button (X) in the top right corner. Inside the dialog, there is a green Excel icon with a gold padlock, followed by the text "My Test Workbook" and "Activation is required before access". Below this, a message reads: "To access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

👉 As chaves de ativação podem ser configuradas com uma [data de expiração](#) ou um [número limitado de execuções](#). Elas também podem ter [bloqueio por hardware](#), o que significa que uma chave só funcionará em um computador específico.

DICA

Além do [Key Generator](#) integrado, oferecemos um gerador de chaves independente capaz de criar milhares de chaves de uma só vez, bem como um SDK de geração de chaves para integração com seu servidor ou site.

Por fim, você pode implementar recursos de [ativação online](#) e [validação](#) para controlar remotamente como os usuários acessam sua planilha e automatizar a entrega das chaves de ativação.



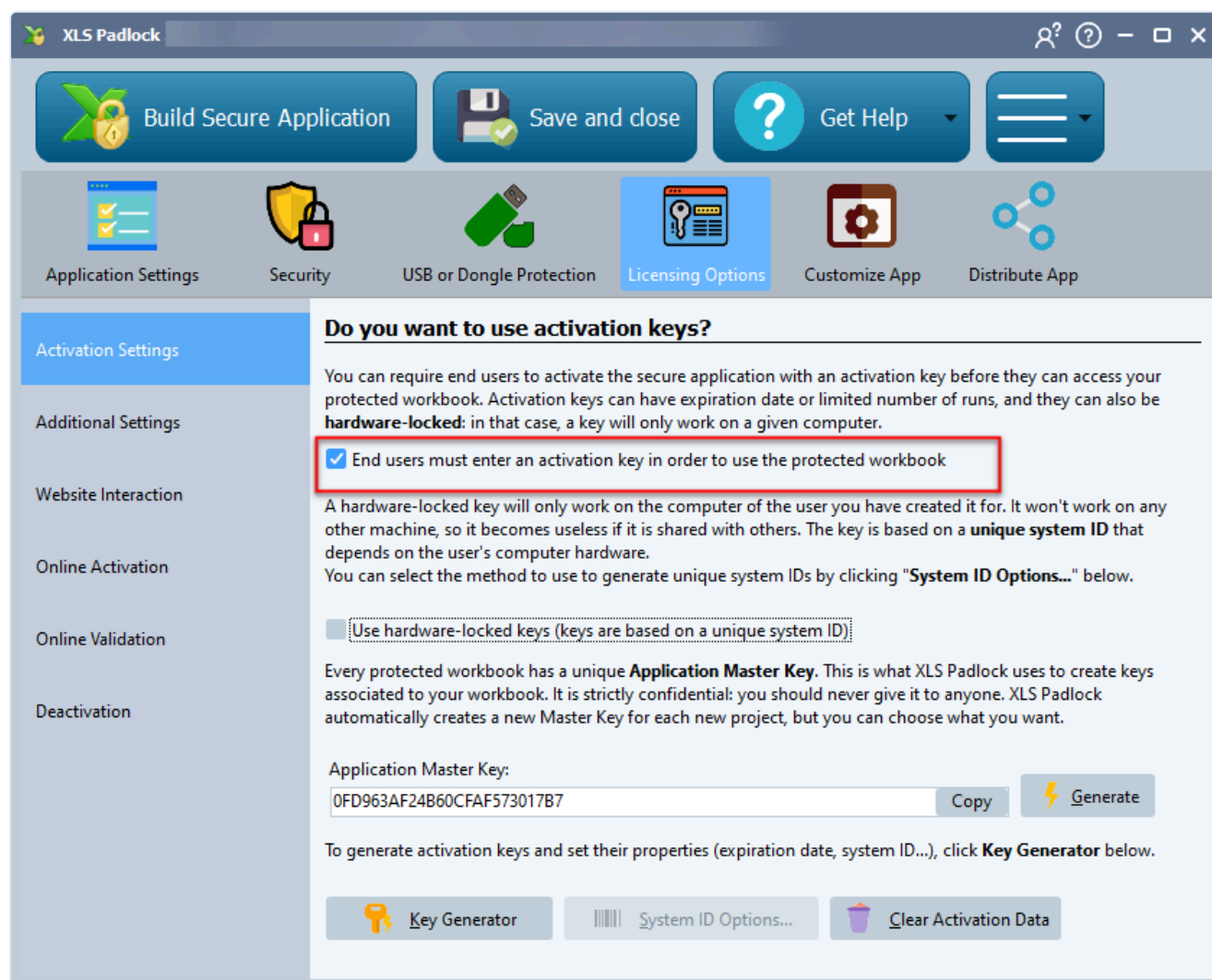
👉 Para começar, consulte nosso tutorial sobre [como configurar as chaves de ativação](#).

Configurar chaves de ativação

As **chaves de ativação** são um poderoso sistema de licenciamento que permite controlar quem pode acessar suas planilhas protegidas. Este guia mostra como começar.

1. Habilitar as chaves de ativação

Nas configurações de segurança do XLS Padlock, marque a caixa **"End users must enter an activation key in order to use the protected workbook"** (os usuários finais devem inserir uma chave de ativação para usar a planilha protegida).



Do you want to use activation keys?

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

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

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

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

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

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

Application Master Key:
0FD963AF24B60CFAF573017B7 Copy Generate

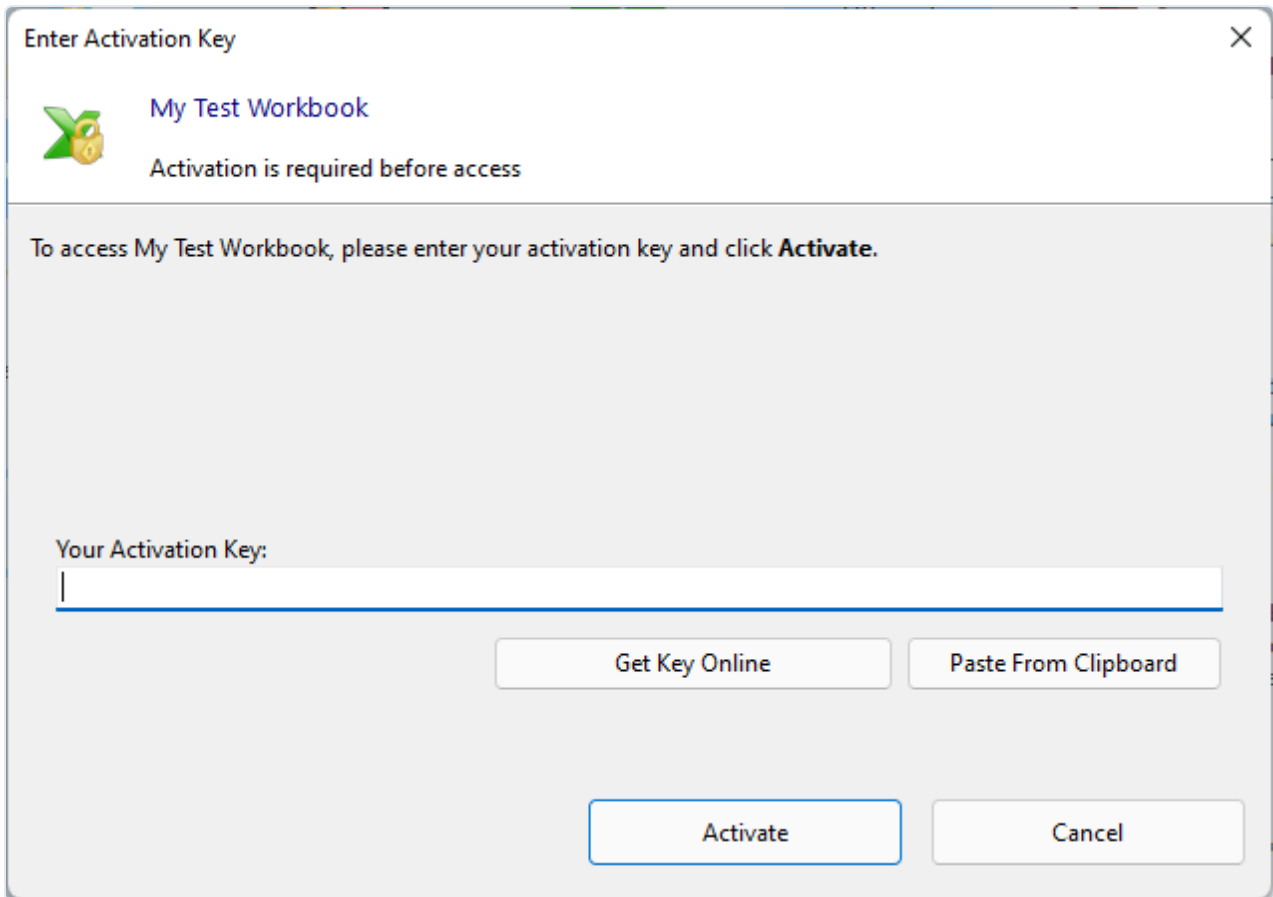
To generate activation keys and set their properties (expiration date, system ID...), click **Key Generator** below.

Key Generator System ID Options... Clear Activation Data

Feito isso, recompile sua aplicação.

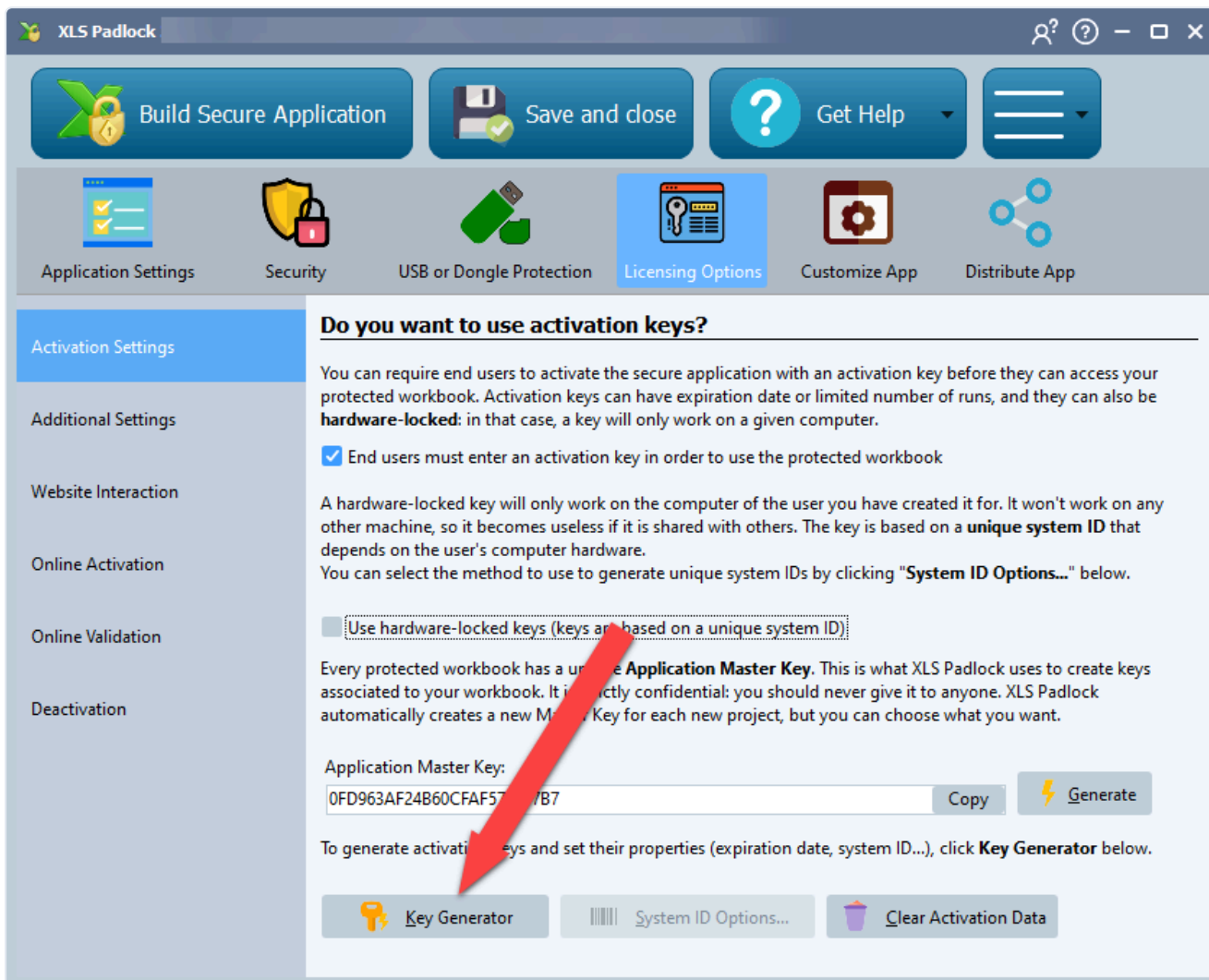
2. Testar sua aplicação

Quando você executar a aplicação recém-compilada, será solicitado a inserir uma chave de ativação antes de poder acessar a planilha:



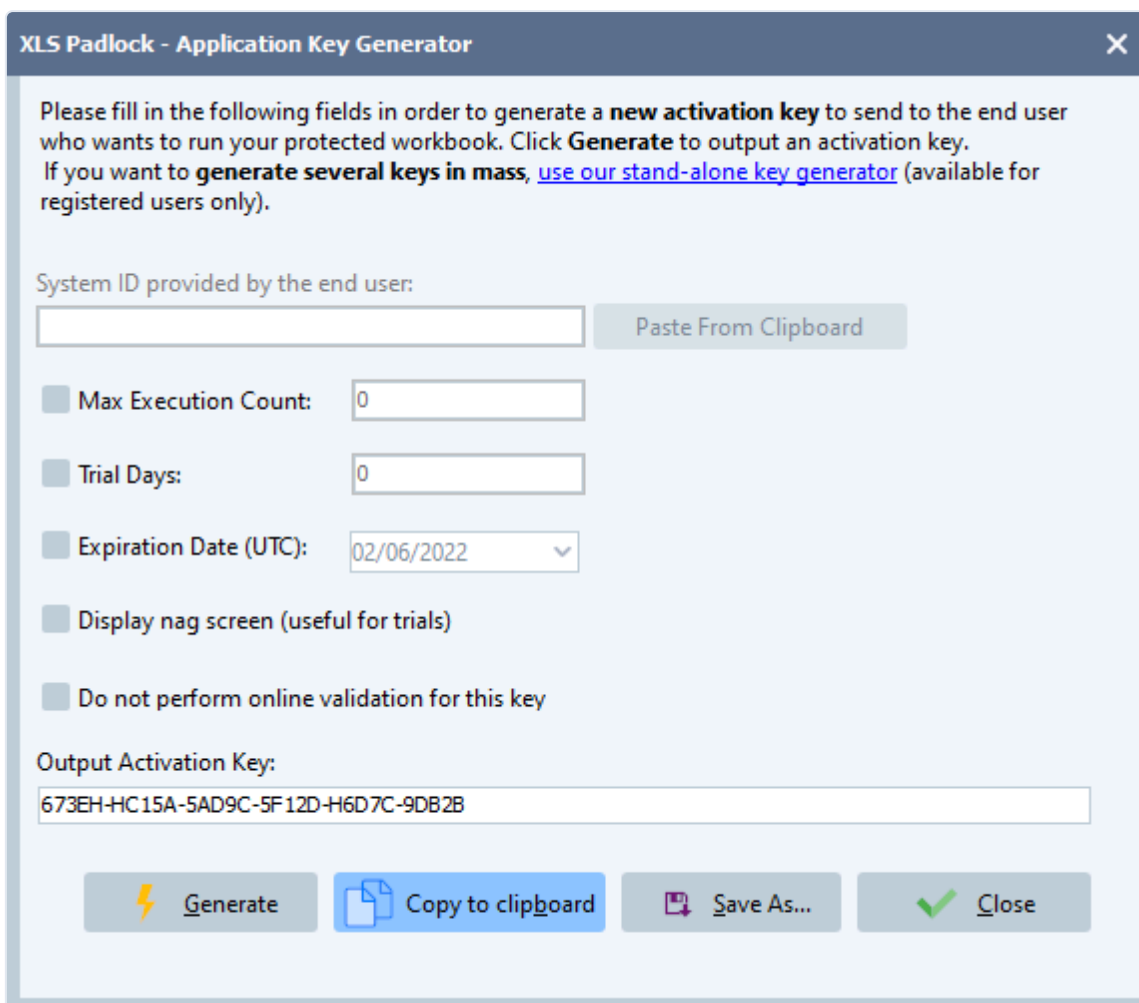
3. Gerar chaves de ativação

Para criar chaves de ativação, abra o [Key Generator](#) (gerador de chaves) no XLS Padlock.

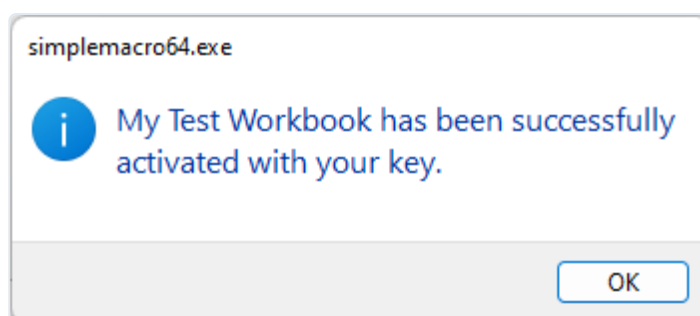


Se desejar, você pode [definir restrições na chave](#), como limitar o número de execuções ou definir uma data de expiração.

Clique no botão **Generate** (gerar) para criar uma chave instantaneamente. Em seguida, você pode copiar essa chave para a área de transferência e enviá-la ao seu cliente.



Quando seu cliente inserir a chave e clicar em **Activate** (ativar), ele receberá acesso à planilha. A aplicação não solicitará a chave novamente, a menos que ela expire.



Automatizar a geração de chaves

Para facilitar a distribuição de chaves, há um aplicativo de [gerador de chaves independente](#) disponível para clientes registrados. Também fornecemos aplicativos web gratuitos para gerar chaves automaticamente por meio do seu site.

Chaves de ativação com bloqueio por hardware

As **chaves de ativação bloqueadas por hardware** são uma forma poderosa de evitar o compartilhamento casual da sua planilha protegida, vinculando cada licença a um computador específico. Este guia mostra como configurá-las.

Tutorial em vídeo disponível

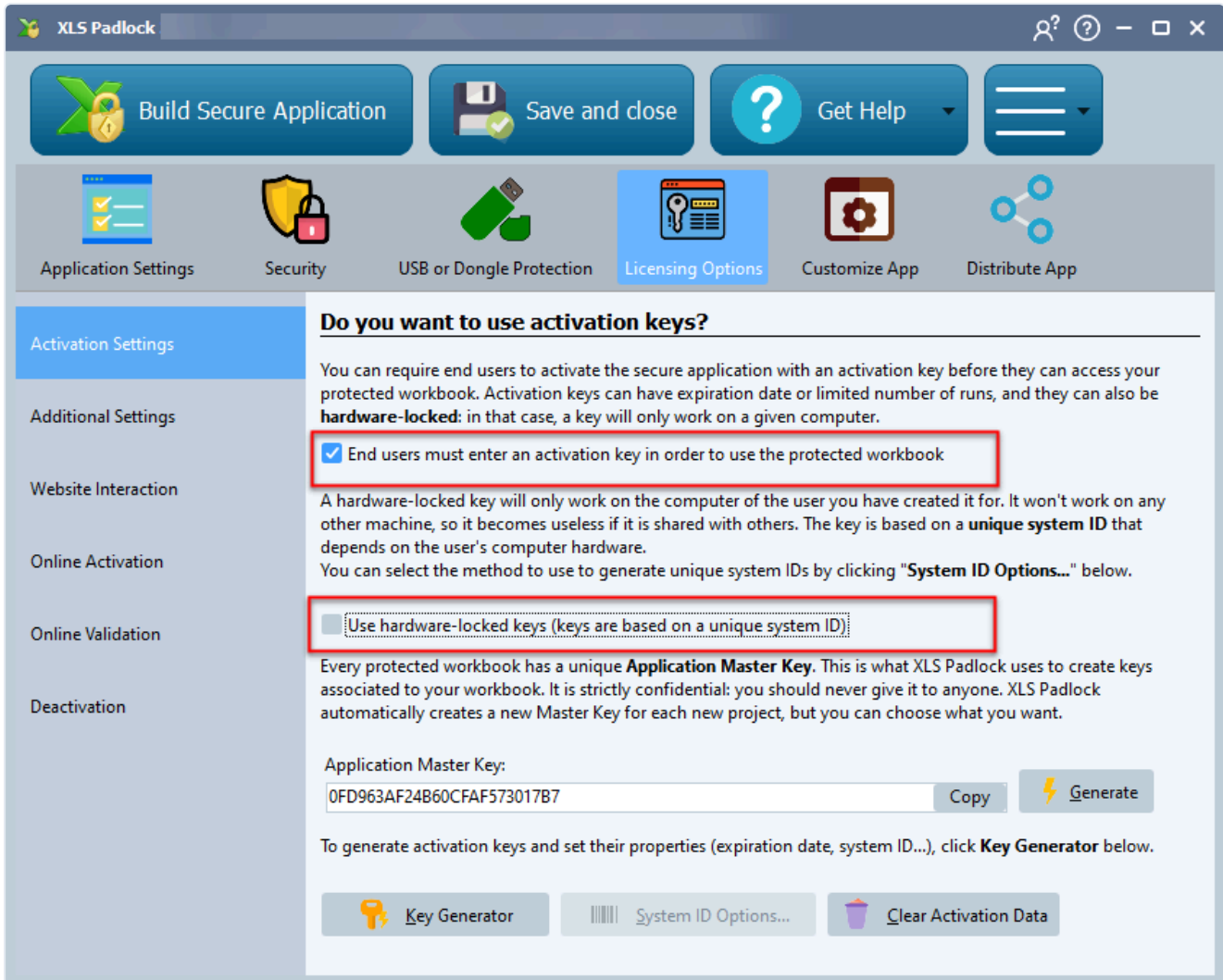
- [Assista ao nosso tutorial em vídeo sobre o bloqueio por hardware das suas planilhas do Excel](#)
- [Veja uma demonstração ao vivo deste recurso](#)

Sobre as chaves bloqueadas por hardware

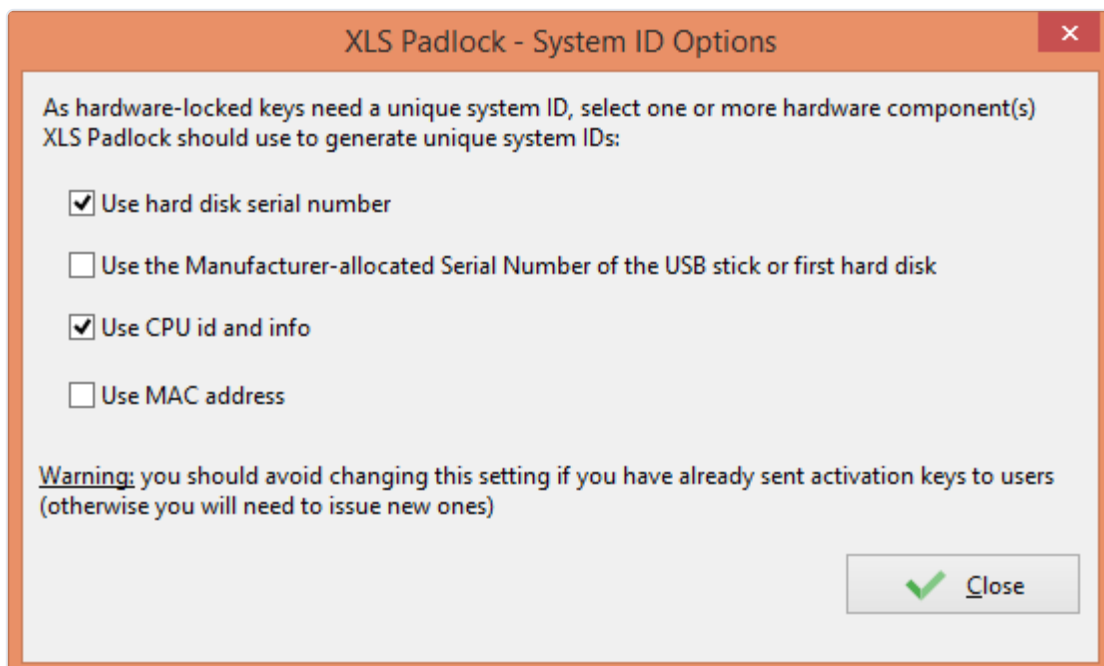
Uma chave bloqueada por hardware **só funcionará no computador para o qual foi gerada**. Ela é baseada em um **System ID** (identificador de sistema) único derivado do hardware do computador do usuário, o que torna a chave inútil se compartilhada com outras pessoas.

1. Habilitar chaves bloqueadas por hardware

Nas configurações do XLS Padlock, marque tanto "End users must enter an activation key..." quanto "**Use hardware-locked keys**" (usar chaves bloqueadas por hardware):



Opcionalmente, você pode configurar quais componentes de hardware são usados para gerar o System ID clicando em "**System ID Options...**" (opções do identificador de sistema). É importante não alterar essas opções depois de ter começado a distribuir sua aplicação.



Notas sobre os componentes do System ID

- **CPU:** Se a CPU não tiver um ID de série integrado, suas informações gerais serão usadas. Em hardware idêntico, isso poderia resultar no mesmo ID. É melhor combinar várias opções de hardware.
- **MAC Address:** Se os usuários tiverem várias formas de se conectar à internet (Wi-Fi, 4G, etc.), o endereço MAC deles pode mudar, o que invalidaria a chave.

Usar a impressão digital de hardware aprimorada

A partir do XLS Padlock 2026.0, a caixa de diálogo System ID Options também expõe uma caixa de seleção adicional: **"Use enhanced hardware fingerprint"** (usar impressão digital de hardware aprimorada).

Quando essa opção está habilitada, o EXE protegido usa um cálculo reforçado do System ID:

- A fonte de bloqueio do Windows lê o identificador único de instalação do Windows diretamente do registro do sistema, em vez da varredura do volume de disco usada pelo algoritmo legado. Esse identificador é estável diante de reformatações de volume e trocas físicas de disco. Apenas uma reinstalação completa do Windows o regenera.
- Todas as fontes de hardware são normalizadas (com espaços aparados e capitalização canonizada) e claramente delimitadas dentro do cálculo da impressão digital, eliminando uma classe de colisões estruturais em que duas máquinas distintas poderiam ocasionalmente chegar ao mesmo System ID. A nova lógica também protege contra falsos negativos quando uma fonte se torna temporariamente indisponível (por exemplo, quando um antivírus bloqueia chamadas WMI), mantendo o System ID estável em vez de alterá-lo.
- O hash subjacente é atualizado para SHA-256 no lugar do antigo MD5, removendo um algoritmo obsoleto da pegada criptográfica do EXE protegido.

O formato exibido do System ID permanece inalterado, ainda com 14 caracteres hexadecimais na forma familiar `XXXX-XXXX-XXXX`, de modo que seus usuários finais e seu fluxo de geração de chaves não notam qualquer diferença.

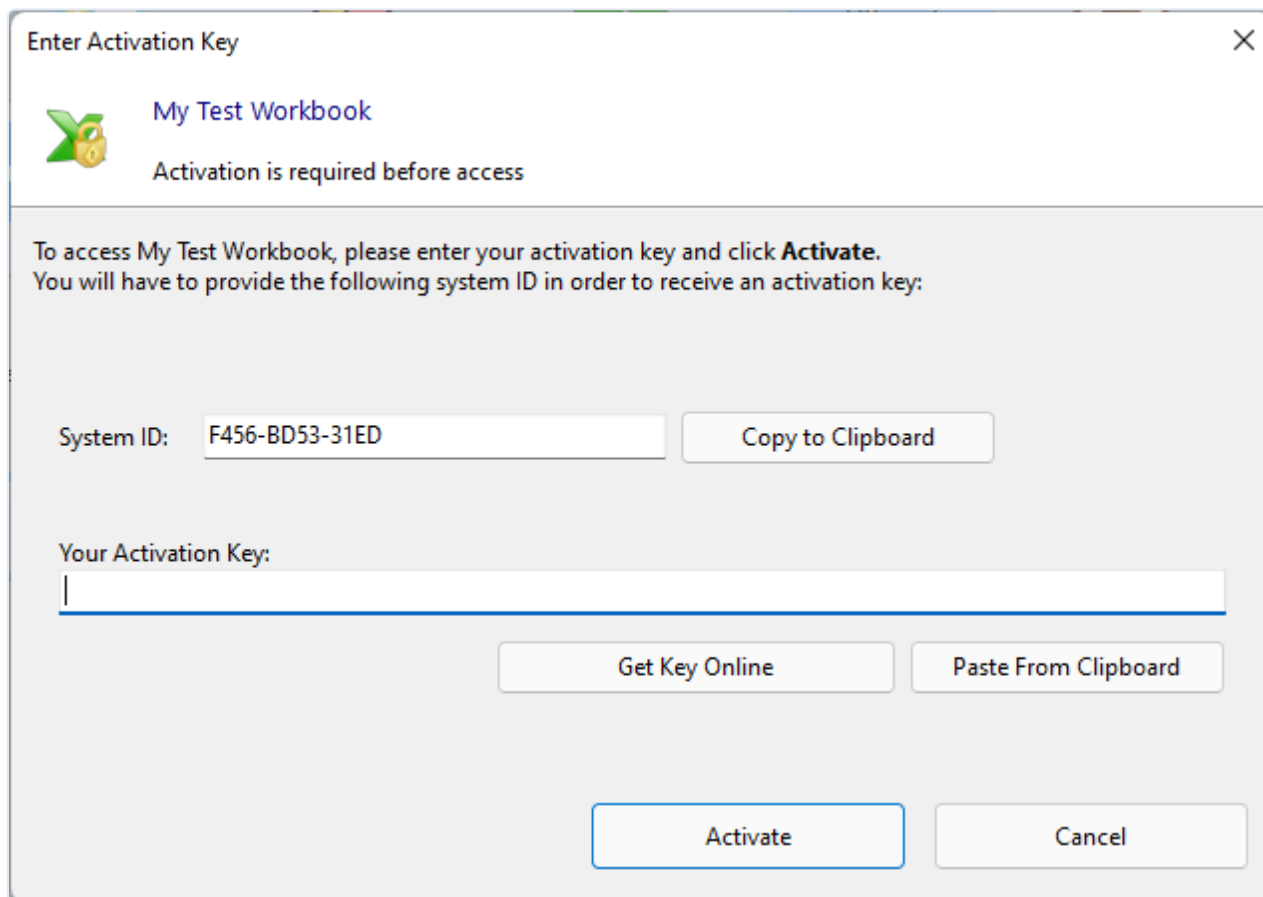
Comportamento padrão:

- **Projetos existentes** (salvos com o XLS Padlock 2025.3 ou anterior): a opção está **DESATIVADA** por padrão. Isso preserva todas as chaves de ativação já em circulação: os System IDs que seus clientes obtiveram em versões mais antigas continuam correspondendo às chaves que você emitiu.
- **Projetos novos** (criados com o XLS Padlock 2026.0 e posteriores): a opção está **ATIVADA** por padrão.

⚠ Importante: alterar essa opção em um projeto que já foi distribuído vai rotacionar o System ID de cada usuário final, invalidando todas as chaves de ativação que você emitiu até o momento. Ative-a apenas em um projeto novo, ou coordene a mudança com uma reemissão da chave de cada cliente.

2. Como os usuários obtêm o System ID

Quando um usuário executa sua aplicação pela primeira vez, será solicitado a inserir uma chave de ativação. O System ID único dele será exibido nessa janela. O usuário deve copiar esse ID e enviá-lo a você.



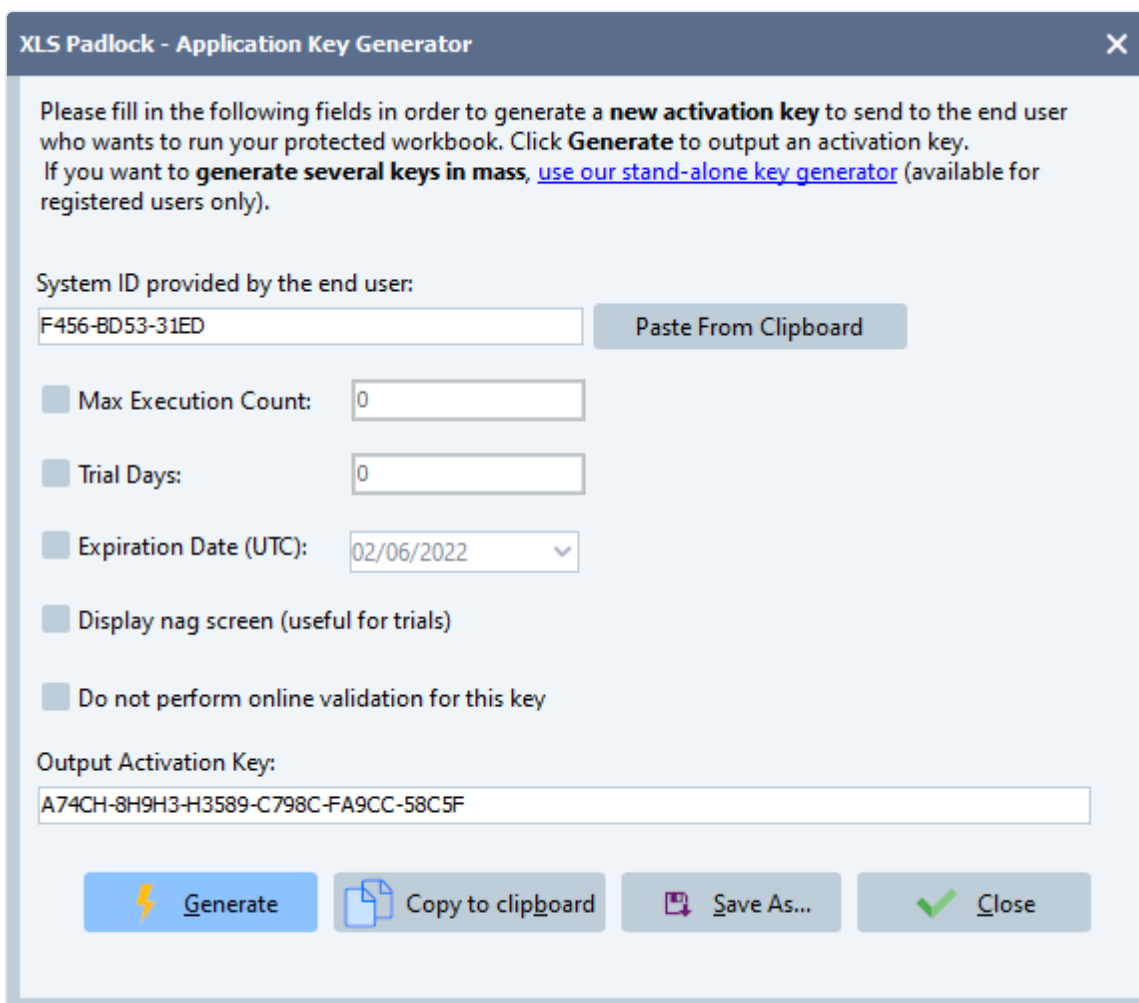
The screenshot shows a dialog box titled "Enter Activation Key" with a close button (X) in the top right corner. Below the title bar, there is a green Excel icon and the text "My Test Workbook" and "Activation is required before access". The main content area contains the following text: "To access My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

Automatizar a entrega de chaves

Para evitar a troca manual de System IDs, você pode configurar um botão "[Get Key Online](#)" que envia o usuário ao seu site com o System ID dele, ou usar o recurso totalmente automatizado de [ativação online](#).

3\ Gerar a chave de ativação

Abra o [Key Generator](#) (gerador de chaves) no XLS Padlock. Cole o System ID do cliente no campo designado e clique em **Generate** (gerar). Em seguida, você pode copiar a chave e enviá-la ao seu cliente.



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:
F456-BD53-31ED Paste From Clipboard

Max Execution Count: 0

Trial Days: 0

Expiration Date (UTC): 02/06/2022 ▾

Display nag screen (useful for trials)

Do not perform online validation for this key

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

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

Quando o cliente inserir a chave, a aplicação será ativada e não solicitará a chave novamente (a menos que ela tenha uma data de expiração). Como a chave está vinculada ao System ID dele, ela não funcionará em nenhum outro computador.

👉 Deseja automatizar a criação de chaves de ativação bloqueadas por hardware? [Leia mais sobre a ativação online.](#)

Ativação on-line

Como funciona

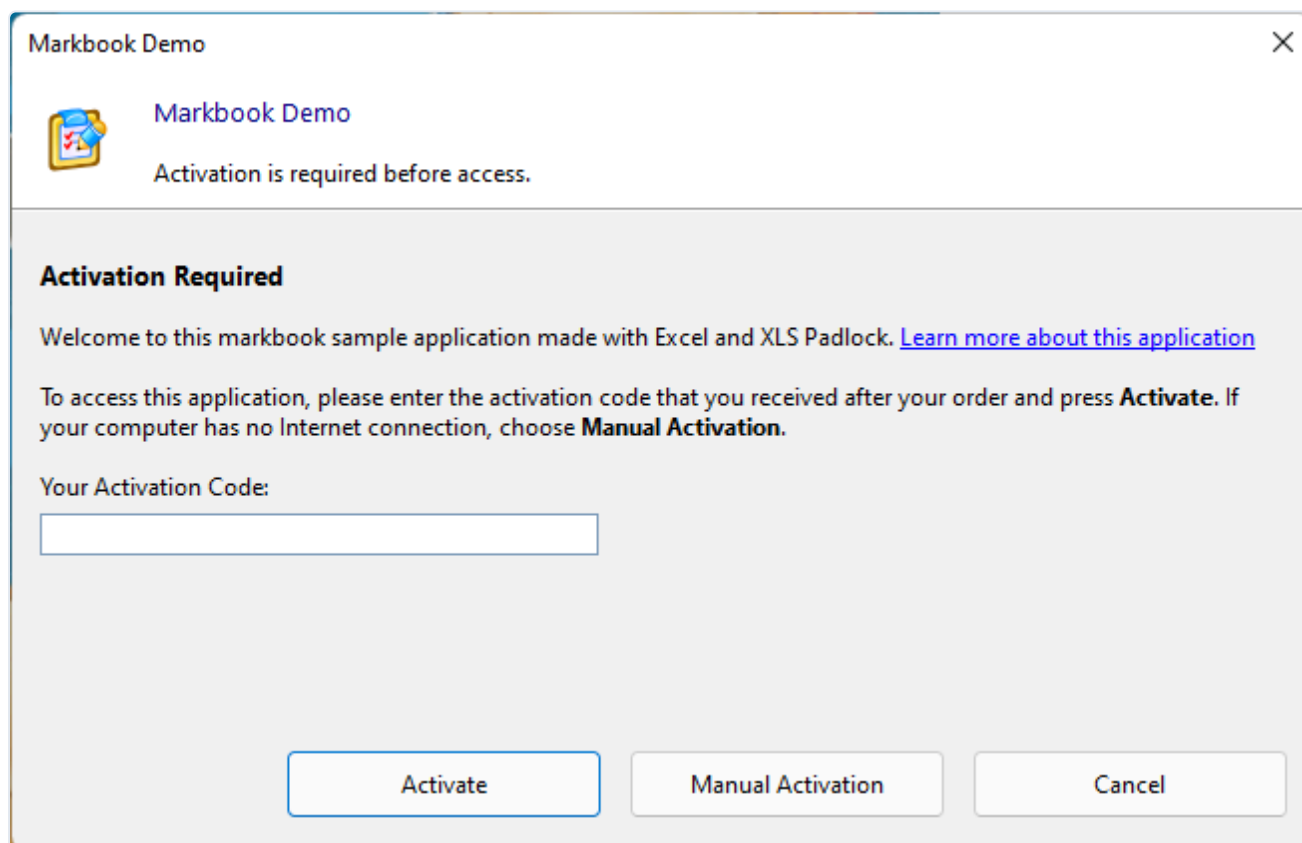
A ativação on-line automatiza a obtenção de **chaves de ativação** pela internet. O aplicativo protegido se comunica com o seu servidor web para baixar uma chave de ativação diretamente, eliminando a necessidade de os usuários a digitarem manualmente.

Kit do lado do servidor obrigatório

Para usar a ativação on-line, você precisa instalar um dos nossos kits do lado do servidor no seu servidor web, como o XLS Padlock Activation Kit, o WooCommerce Integration Kit ou o FastSpring Subscription kit. Você pode baixar esses kits na sua [página de conta](#).

Se você optar pelo protocolo de ativação 2026 (assinatura de respostas Ed25519, veja abaixo), certifique-se de que o seu kit seja a versão 2026.0 ou posterior. O kit detecta automaticamente o protocolo em cada solicitação, de modo que as planilhas legadas continuam funcionando com a mesma instalação do kit.

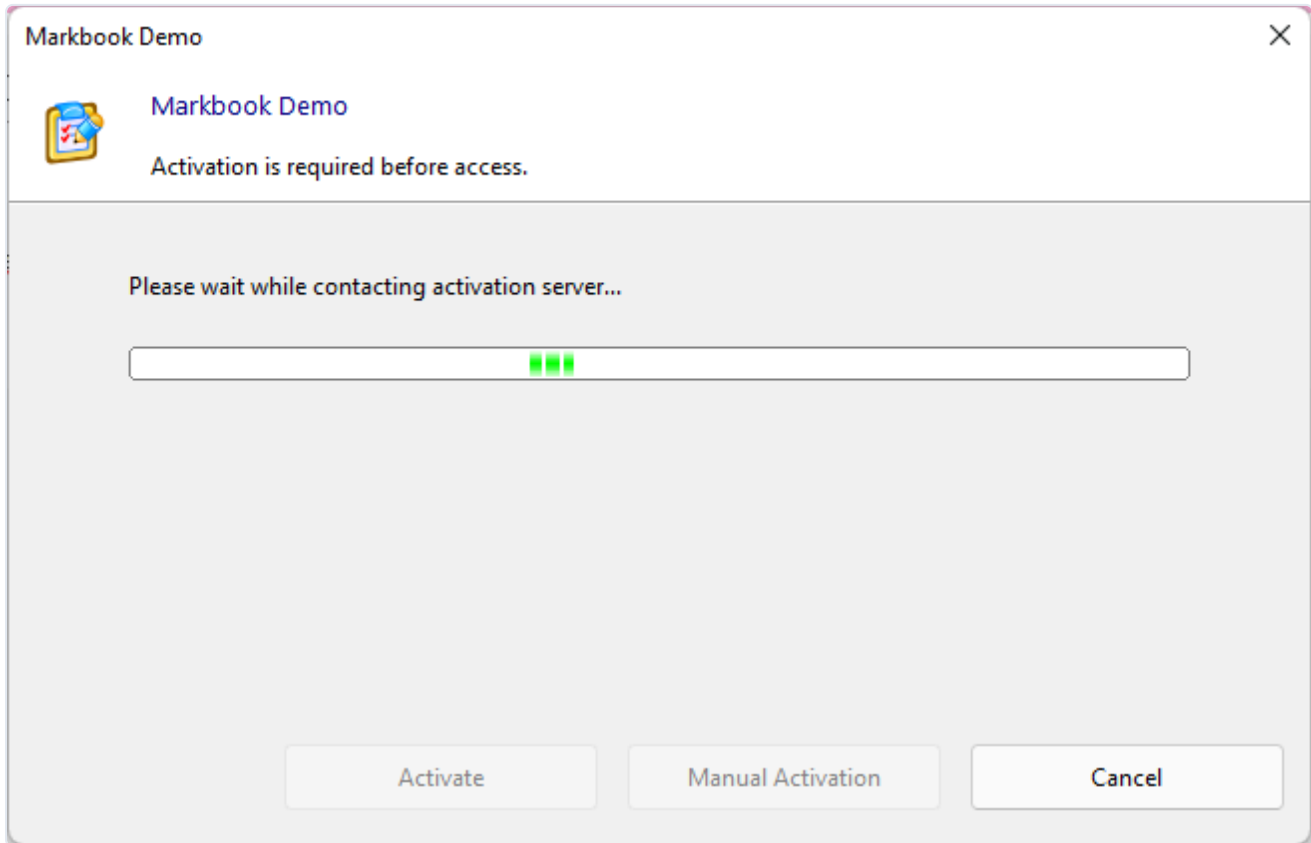
👉 Quando um usuário final inicia o aplicativo, aparece uma caixa de diálogo solicitando a ativação. Essa caixa de diálogo substitui a caixa de diálogo padrão "Enter Activation Key" e é totalmente personalizável. Você pode adicionar os seus próprios campos para coletar dados do usuário (como um número de pedido ou um endereço de e-mail), que serão enviados ao seu servidor web. Em seguida, o seu servidor valida essas informações e, se obtiver êxito, envia uma chave de ativação de volta ao aplicativo.



The screenshot shows a dialog box titled "Markbook Demo" with a close button (X) in the top right corner. The dialog contains the following text and elements:

- Icon of a book with a checkmark and a key.
- Text: "Markbook Demo"
- Text: "Activation is required before access."
- Section header: "Activation Required"
- Text: "Welcome to this markbook sample application made with Excel and XLS Padlock. [Learn more about this application](#)"
- Text: "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**."
- Text: "Your Activation Code:"
- Text input field for the activation code.
- Buttons: "Activate", "Manual Activation", and "Cancel".

Quando o usuário clica em Activate, os dados são enviados ao servidor de ativação:



Após uma ativação bem-sucedida, uma mensagem de confirmação é exibida e o aplicativo é reiniciado. Se ocorrer um erro, uma caixa de mensagem aparecerá, permitindo que o usuário tente novamente.

Configuração

👉 Para habilitar a ativação on-line, você precisa configurar as seguintes opções:

Base Activation URL

Informe a URL completa do kit de ativação instalado no seu servidor. Por exemplo, se você instalou o kit em uma subpasta chamada "activation", a URL seria

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

Use HTTPS

Conexões seguras com TLS/SSL são suportadas. Você deve sempre usar URLs que comecem com `https://` .

⚠️ Deixe o campo em branco se você ****não**** quiser usar a ativação on-line.

Identificação do cliente: Security Key ou par de chaves Ed25519

A partir do XLS Padlock 2026.0, você pode escolher entre dois regimes de protocolo de ativação por meio da opção **Compatibility mode for pre-2026 activation kits** (modo de compatibilidade para kits de ativação anteriores a 2026):

- **Compatibility mode marcado** (padrão, recomendado se o seu servidor de ativação for anterior à versão 2026): o campo GUID legado **Security Private Key** identifica o seu aplicativo perante o kit de ativação. O protocolo é baseado em linhas com respostas sem assinatura. Este é o comportamento de todas as versões do XLS Padlock anteriores a 2026.0.
- **Compatibility mode desmarcado** (requer a versão 2026 do XLS Padlock Activation Kit, do WooCommerce Integration Kit ou do FastSpring Subscription Kit): o campo legado fica oculto e um **par de chaves Ed25519** por projeto assume o controle. Clique no botão **Generate keypair** (gerar par de chaves) na página Online Activation para produzir um novo par de chaves. A chave pública é incorporada na planilha protegida; a chave privada é exibida a você em uma caixa de diálogo de exibição única para que você possa colá-la na configuração do seu kit de ativação (`xlspadlocksignkey` no `config.ini`).

Se você perder a chave privada, clique em **Show config.ini snippet** (mostrar trecho do config.ini) na mesma página para exibi-la novamente. A chave é mantida no seu arquivo de projeto `.xplp`. Trate o `.xplp` como informação confidencial: não o inclua no controle de versão, não o envie por e-mail para suporte.

O protocolo 2026 usa um envelope de solicitação JSON e verifica uma assinatura Ed25519 destacada em cada resposta do servidor de ativação. Isso protege os seus clientes contra respostas falsas de "ativado" ou mensagens de erro de phishing que um invasor no nível da rede poderia, de outra forma, injetar contornando o TLS (proxy corporativo com injeção de CA, antivírus malicioso, certificado raiz comprometido).

Allow Manual Activation if No Internet Connection

Alguns usuários podem não ter uma conexão ativa com a internet. Para permitir que eles ativem manualmente, habilite a opção "**Allow Manual Activation if No Internet Connection**" (permitir ativação manual se não houver conexão com a internet).

Esse método manual funciona da mesma forma que as chaves de registro padrão. Nesse caso, você precisa estar preparado para lidar com as solicitações de ativação de usuários que estejam off-line.

Aviso

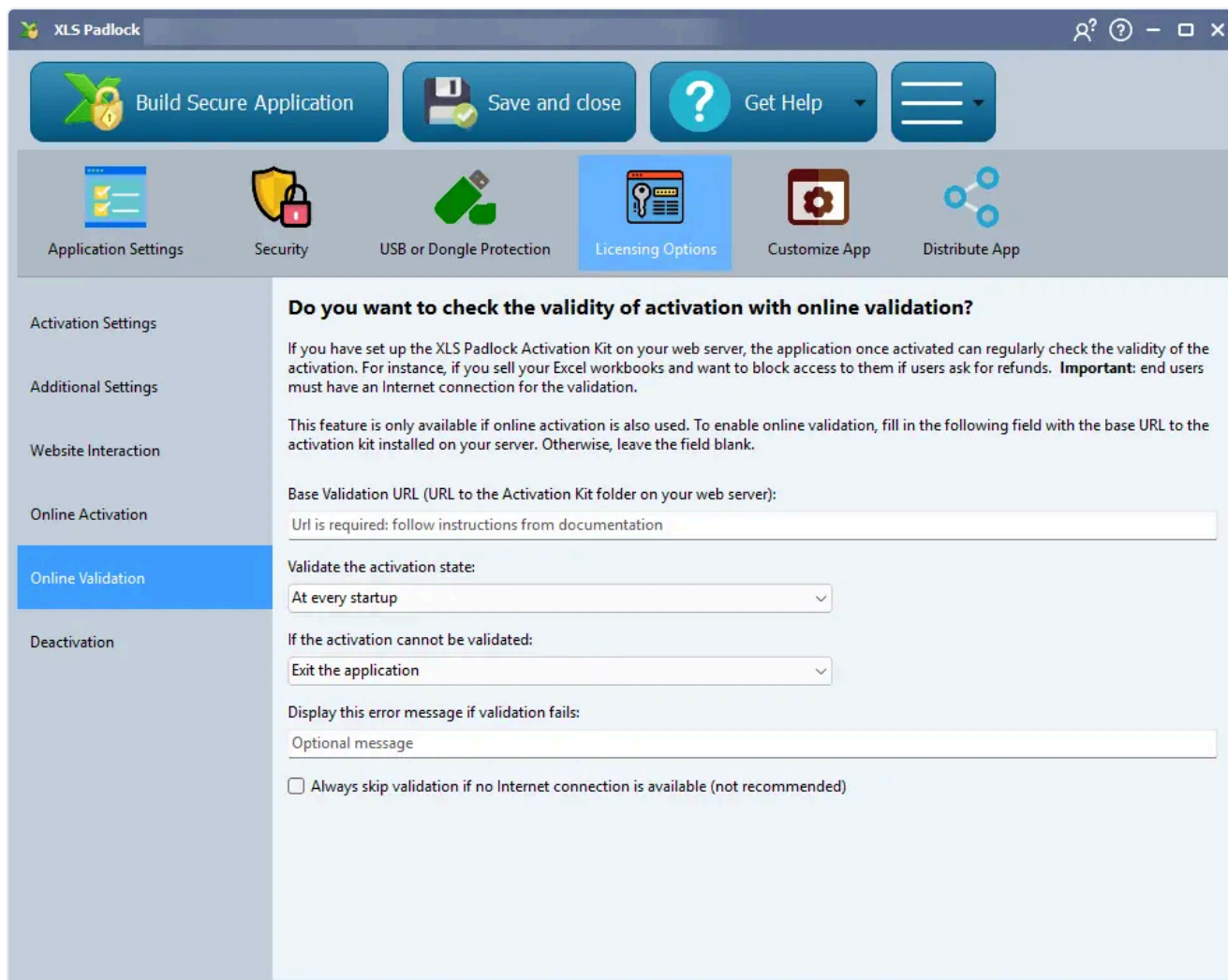
Se você aceitar ativações manuais, certifique-se de gerar chaves de ativação que **não realizem validação on-line** (opção no [gerador de chaves](#)). Caso contrário, a validação falhará!

Veja também

- [Como vender planilhas do Excel com assinaturas](#)
- [Mostrar um botão "Purchase Online" na tela de aviso](#)

Validação on-line

A validação on-line proporciona **controle remoto** sobre os seus aplicativos ativados. Quando usada com um dos nossos kits de ativação do lado do servidor (como o XLS Padlock Activation Kit, o WooCommerce Kit ou o [FastSpring subscription kit](#)), o seu aplicativo pode verificar periodicamente com o seu servidor se a sua ativação continua válida.



Aviso

Este recurso requer que a [ativação on-line](#) esteja habilitada e configurada previamente.

Isso é útil em vários cenários:

- Você pode desativar remotamente um aplicativo se um cliente solicitar um reembolso.
- Você pode impor um acesso baseado em assinatura, no qual o aplicativo deixa de funcionar se uma assinatura expirar.

Como funciona

- A validação on-line usa o mesmo token único que foi gerado durante a ativação on-line inicial para identificar o dispositivo do cliente.
- A chave de ativação em si nunca é enviada pela internet durante a validação.
- Uma conexão ativa com a internet é necessária para que a verificação de validação ocorra.

Configuração

👉 Você precisa configurar a **Base Validation URL** para apontar para o seu script de validação do lado do servidor. Normalmente, essa é a URL do XLS Padlock Activation Kit, do XLS Padlock WooCommerce Integration Kit ou do FastSpring subscription kit no seu servidor web. Por exemplo, se você instalou o kit de ativação em uma subpasta chamada "activation", a URL seria

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

Use HTTPS

Conexões seguras com TLS/SSL são suportadas. Você deve sempre usar URLs que comecem com `https://` .

⚠️ Deixe o campo em branco se você ****não**** quiser usar a validação on-line.

Frequência de validação

Escolha quando o aplicativo deve realizar uma validação: a cada inicialização, de forma aleatória, a cada X dias ou a cada X execuções. Você precisa especificar o valor de X onde for necessário.

Aviso

Após a ativação, uma validação inicial é necessária na próxima inicialização. Após essa primeira verificação, o aplicativo seguirá a frequência de validação selecionada.

Ação em caso de falha de validação

Defina o que o aplicativo deve fazer se a validação on-line falhar: * **Exit the application:** o aplicativo se fechará imediatamente. * **Blacklist activation key:** a chave atual será invalidada e o usuário será solicitado a digitar uma nova. Se o usuário digitar novamente a mesma chave, a validação será tentada outra vez. * **Do nothing:** o aplicativo continuará em execução. Você pode usar as extensões VBA do XLS Padlock para verificar o status da validação e implementar uma lógica personalizada.

Se a ativação não puder ser validada

Você pode exibir uma mensagem de erro personalizada ao usuário se a validação falhar, orientando-o sobre o que fazer em seguida.

Omitir a validação se estiver off-line

Você pode permitir que o aplicativo omita a verificação de validação se nenhuma conexão com a internet for detectada. Em geral, isso não é recomendado por motivos de segurança.

Aviso

Se esta opção estiver desabilitada, os usuários precisam ter uma conexão ativa com a internet para o processo de validação. Caso contrário, a validação falhará.

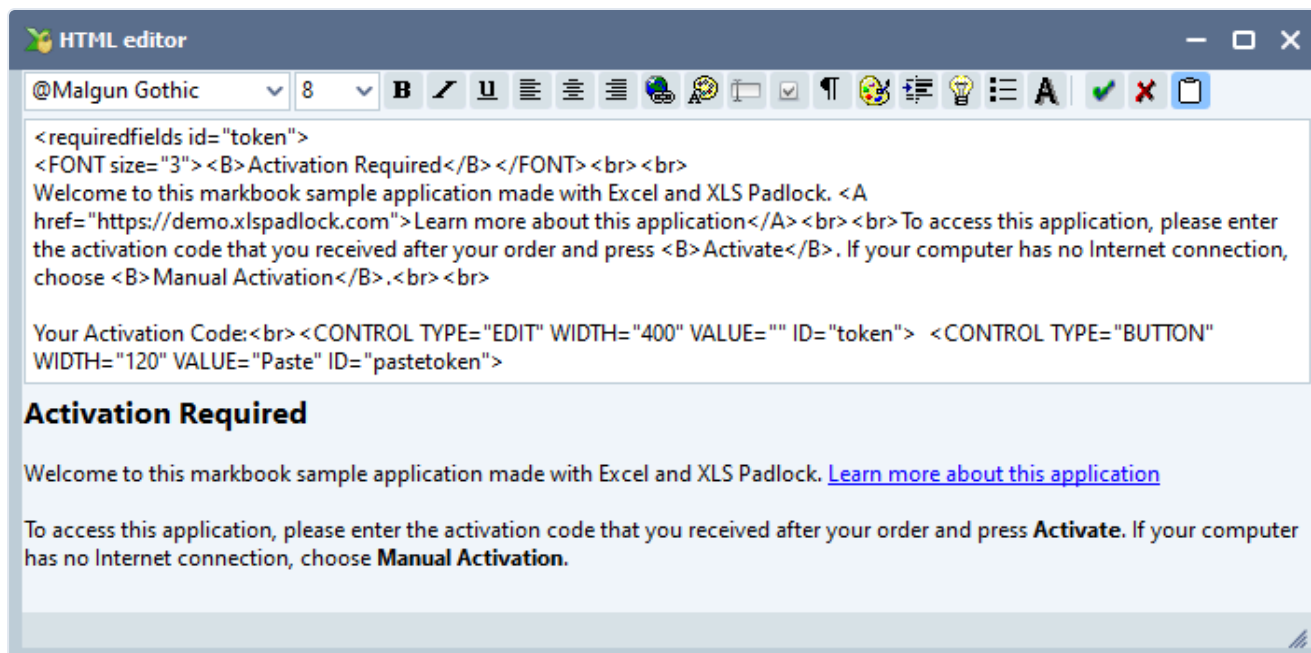
VBA: Recuperar informações de assinatura/licença

Após uma validação on-line bem-sucedida, você pode recuperar dados personalizados enviados pelo seu servidor usando a chamada VBA `XLSPadLock.PLEvalVar("ValidationAddServerData")`. Isso é útil para repassar informações específicas de assinatura para a planilha.

Editor do formulário de registro

O Editor do formulário de registro permite que você personalize o texto que aparece na [caixa de diálogo de ativação on-line](#). O editor oferece suporte a HTML básico, então você pode usar tags como ``, `
`, ``, etc.

O XLS Padlock fornece um editor HTML básico com uma pré-visualização ao vivo do texto da caixa de diálogo:



Controles personalizados

Você pode adicionar campos personalizados para solicitar informações adicionais aos usuários, que serão enviadas ao seu servidor web. Você também pode adicionar um botão para colar texto no campo do token.

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

Campos obrigatórios

Você pode tornar os campos obrigatórios indicando seus IDs em uma tag `<requiredfields>`. Por exemplo, para exigir os campos "token" e "name":

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

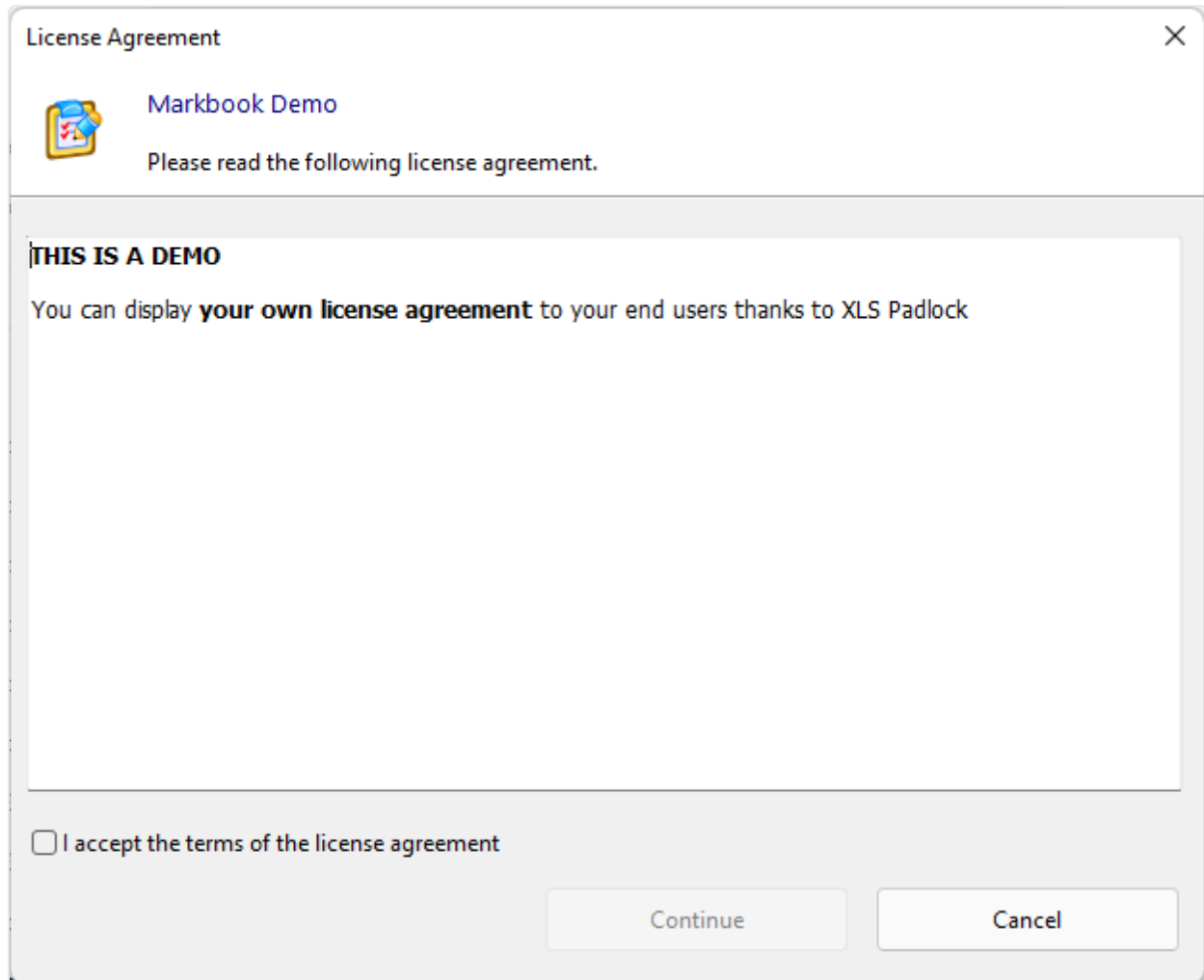
Exemplo completo

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

Mostrar o contrato de licença

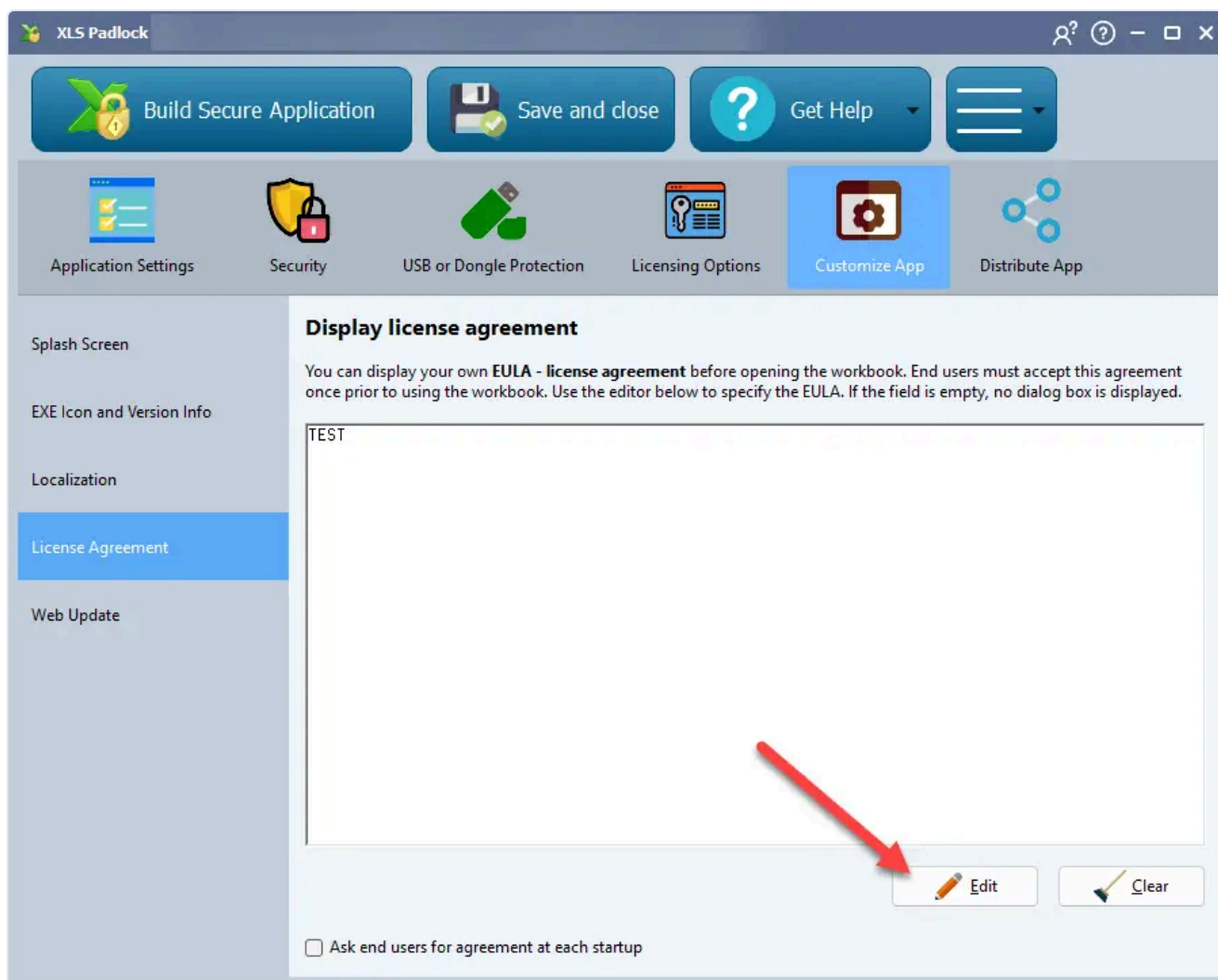
Você pode exigir que os usuários aceitem um **acordo de licença de usuário final (EULA)** antes de poderem abrir sua planilha protegida.

A planilha compilada exibirá uma caixa de diálogo como a mostrada abaixo:



Quando o usuário marca "I accept the terms of the license agreement", o botão "Continue" fica ativo, permitindo que ele prossiga.

👉 Para adicionar seu EULA, clique em **Edit** (Editar) para abrir um editor de texto formatado nesta página:



Você pode carregar conteúdo a partir de um arquivo RTF ou colar texto diretamente de um aplicativo como o Microsoft Word.

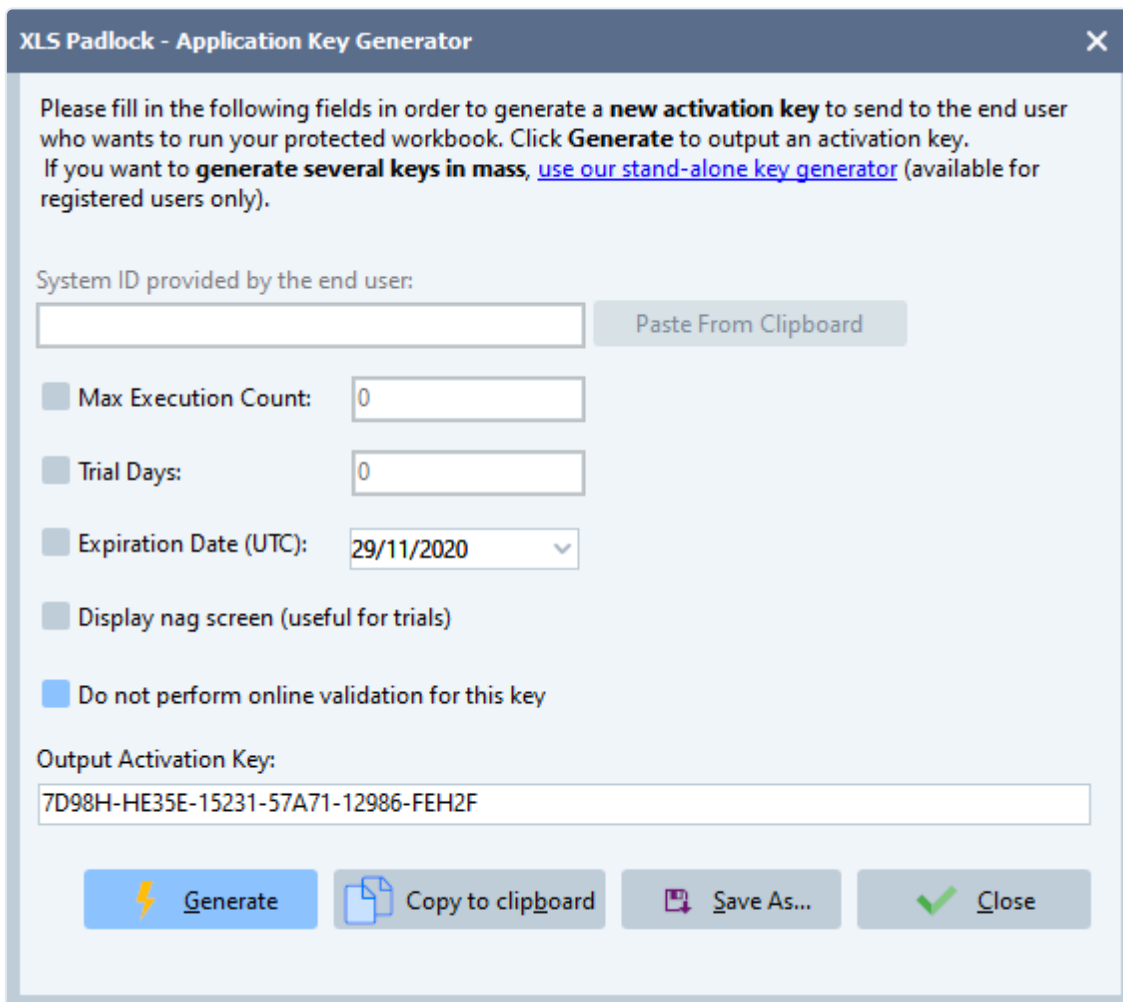
OBSERVAÇÃO

Se o conteúdo do EULA for deixado vazio, nenhuma caixa de diálogo será exibida.

i Para exibir o EULA toda vez que o aplicativo é iniciado, habilite a opção **Ask end users for agreement at each startup** (pedir aceitação aos usuários finais a cada inicialização). Por padrão, os usuários só precisam aceitá-lo uma vez.

Gerador de chaves (portátil e servidor)

O XLS Padlock inclui um **Key Generator** integrado para criar chaves de ativação para suas aplicações de planilha.



- **Para chaves de ativação simples**, basta clicar no botão **Generate** (gerar) para criar uma chave instantaneamente.
- **Para chaves vinculadas ao hardware**, informe o System ID fornecido pelo seu usuário e, em seguida, clique em **Generate**. A chave resultante ficará vinculada a esse computador específico.

Você pode copiar a chave gerada para a área de transferência a fim de enviá-la ao seu cliente, ou salvá-la como um arquivo `.txt` para seus registros.

As chaves são específicas de cada planilha

O gerador de chaves cria chaves de ativação apenas para o projeto de planilha aberto no momento. As chaves geradas para uma planilha não funcionarão com outra, a menos que compartilhem a mesma [Application Master Key](#).

SDK do gerador de chaves

Oferecemos um SDK do Key Generator gratuito (uma aplicação web baseada em PHP) que você pode instalar no seu próprio site. Este SDK permite automatizar a geração e a entrega de chaves de ativação para suas planilhas.

Diferentemente do [Key Generator integrado](#), que cria chaves manualmente dentro do XLS Padlock, este SDK é executado no seu próprio servidor para emitir chaves de forma automática.

Para uma solução de licenciamento totalmente automatizada, considere combinar o SDK com o [recurso de Ativação on-line](#).

👉 Você pode baixar o SDK do Key Generator na sua [página de conta](#).

Gerador de chaves independente

O XLS Padlock oferece um gerador de chaves independente para clientes registrados. Essa ferramenta permite que você [crie chaves de ativação](#) para suas planilhas protegidas sem precisar abrir o Excel ou o suplemento do XLS Padlock.

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

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

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

Application Master Key:
D... Paste From Clipboard

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

System ID provided by the end user:
Paste From Clipboard

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

Max Execution Count: 0

Trial Days: 0

Expiration Date (UTC): 19/12/2016

Display nag screen

Generate Key(s) Number of Keys: 100

Load/Save Profile: Save Load

Copy to clipboard Save As... Close

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

Para um gerenciamento mais simples, o gerador de chaves independente também permite salvar configurações como **profiles** (perfis).

👉 Você pode baixar o gerador de chaves independente na sua [página de conta](#).

Restrições nas chaves

Ao gerar [chaves de ativação](#) com o XLS Padlock, você pode **aplicar várias restrições** usando as seguintes opções:

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)

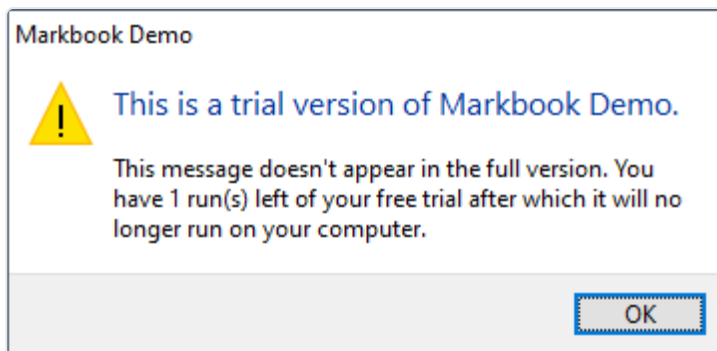
Do not perform online validation for this key

Output Activation Key:

⚡ Limite quantas vezes seu aplicativo pode ser executado habilitando **"Max Execution Count"** (número máximo de execuções) e informando o número desejado.

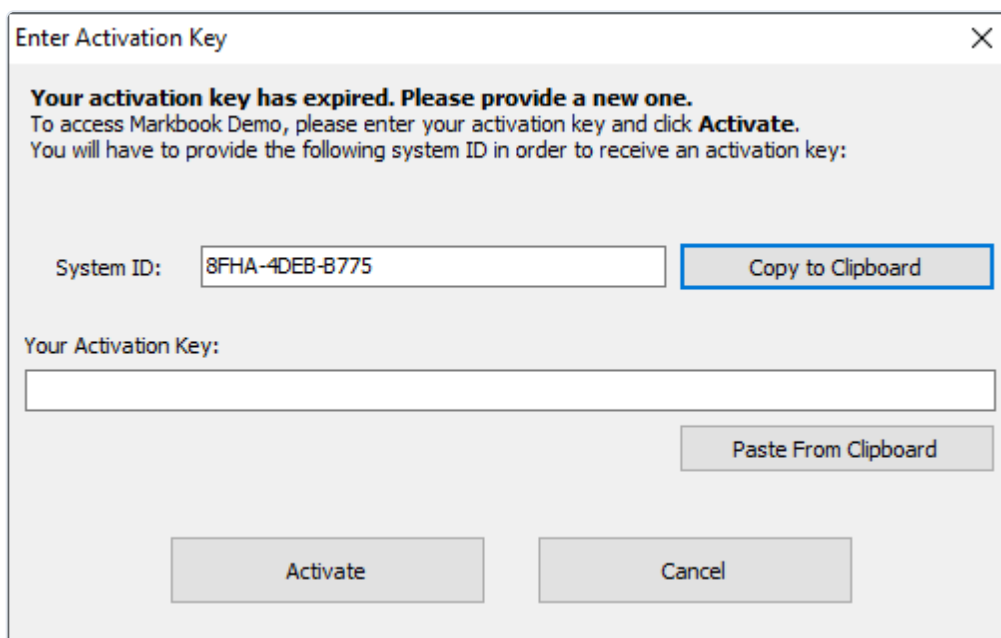
⚡ Defina a chave para expirar após um número específico de dias (habilite **"Trial Days"**, dias de avaliação) ou em uma data específica (habilite **"Expiration Date"**, data de expiração).

⚡ Exiba uma caixa de diálogo de lembrete (ou "nag screen") na inicialização habilitando **"Display Nag Screen"**. Isso é [útil para versões de avaliação](#), pois informa ao usuário os dias ou as execuções restantes.



Por fim, se você usar a [validação online](#) e precisar isentar uma chave específica dessa verificação (por exemplo, para um usuário em um ambiente sem conexão), pode habilitar "**Do not perform online validation for this key**" (não realizar validação online para esta chave). Isso só é recomendado quando usado em combinação com chaves vinculadas por hardware.

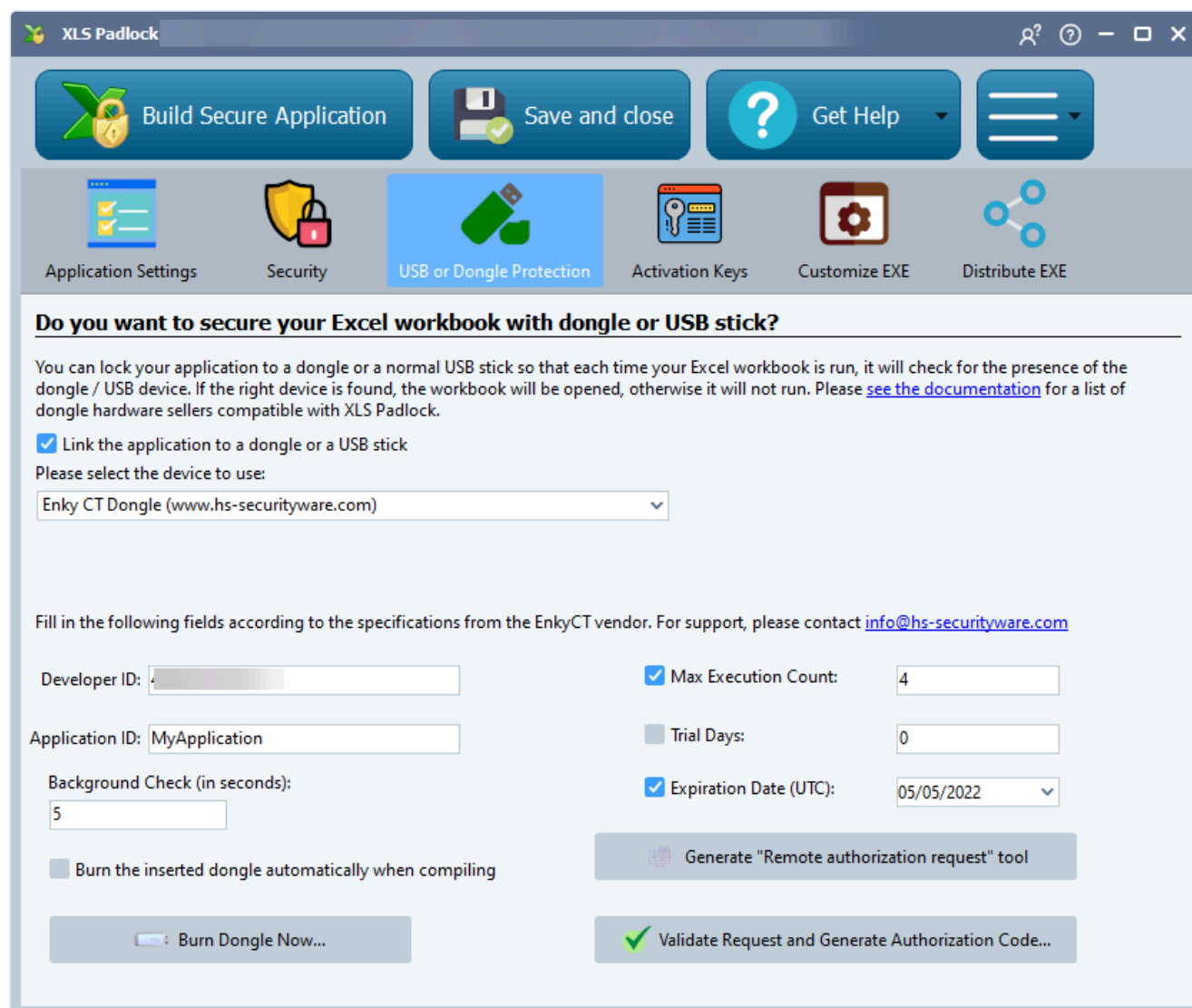
Quando o limite de uma chave (data, número de execuções ou dias de avaliação) é atingido, o usuário será solicitado a inserir uma nova chave para continuar usando o aplicativo:



Proteção por USB ou dongle

Com o XLS Padlock, você pode vincular suas planilhas do Excel a um pen drive USB ou a um dongle de segurança dedicado. Isso significa que **o aplicativo da planilha compilada não funcionará se o dispositivo USB correto não estiver inserido**. O aplicativo verificará a presença do dispositivo na inicialização e também pode verificá-la periodicamente durante a execução.

Para usar essa proteção, marque a opção **"Link the application to a dongle or a USB stick"** (vincular o aplicativo a um dongle ou pen drive USB) na página "USB Protection" e selecione o modelo de dispositivo que você possui.



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

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

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

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

Fields and options include:

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

O XLS Padlock oferece suporte a vários tipos de proteção baseada em hardware para vincular seu aplicativo do Excel a um dispositivo físico.

Dongle Enky CT

Segundo o fornecedor, o Enky CT é um dongle de relógio de ponto baseado em cartão inteligente de 32 bits, fácil de usar e flexível, utilizado principalmente para proteção de software e limitação de tempo.

Para obter suporte, entre em contato com info@hs-securityware.com ou visite <https://hs-securityware.com>.

Como usar o dongle Enky CT

Insira o **Developer ID** fornecido pela HS Securityware (diferencia maiúsculas de minúsculas).

O **Application ID** pode ser qualquer valor que você queira para identificar o dongle, e deve estar relacionado ao aplicativo que você está criando.

Background Check

O aplicativo verifica a presença do dongle na inicialização e também pode verificá-la periodicamente durante a execução. Você pode definir a frequência dessas verificações; por exemplo, um valor de 20 significa que o aplicativo verificará o dongle a cada 20 segundos. Se o dongle não for encontrado, uma tela de aviso aparece, exigindo que o usuário insira o dongle dentro de 15 segundos, ou o aplicativo será encerrado.

Set restrictions on the application

Você pode limitar o número de vezes que seu aplicativo pode ser executado definindo um **Max Execution Count**. Você também pode fazer com que seu aplicativo expire após um número determinado de dias (**Trial Days**) ou em uma **Expiration Date** específica.

Recursos de atualização remota de dongles

Esse modelo de dongle permite que você atualize remotamente os dongles que já estão em poder dos seus clientes. Por exemplo, você pode estender uma data de expiração ou adicionar mais execuções. Esse recurso de atualização remota baseia-se em um sistema de código de solicitação e autorização.

Como atualizar remotamente o dongle de um cliente:

1. Primeiro, gere a "remote authorization request tool" (ferramenta de solicitação de autorização remota) a partir do XLS Padlock e envie ao seu cliente os arquivos EXE e DAT resultantes.

Aviso

Esses dois arquivos (`.EXE` e `.DAT`) devem permanecer na mesma pasta para que a ferramenta funcione.

2. O cliente executa a ferramenta para gerar um **request code** (código de solicitação), que ele envia para você.
 3. No XLS Padlock, escolha "Validate Request and Generate Authorization Code", cole o código de solicitação e clique em "Validate Request Code". Em seguida, você poderá ver o status atual do dongle e definir novos limites (número de usos ou uma nova data de expiração).
 4. Gere o **authorization code** (código de autorização) e envie-o de volta ao cliente. Ele usará a mesma ferramenta para inserir o código e aplicar a atualização ao seu dongle.
-

Dongle Enky LC

Para obter suporte, entre em contato com info@hs-securityware.com ou visite <https://hs-securityware.com>.

O Enky LC é um dongle HID econômico e sem drivers para proteção de software.

Como usar o dongle Enky LC2

1. Insira o **Developer ID** que você recebeu da HS-Security Ware.
2. Insira um **Product ID** único para sua planilha. Isso garante que apenas dongles com o Product ID correto serão aceitos.

Etapas para configurar um dongle Enky LC2

Para vincular um dongle ao seu aplicativo, primeiro você deve "gravar" (burn) o dongle usando o botão "**Burn Dongle Now**" no XLS Padlock. Isso configura o dongle com seus identificadores. Esse processo também pode ocorrer automaticamente quando você compila seu aplicativo se um dongle compatível estiver conectado.

Background Check

Você pode ativar uma verificação em segundo plano para garantir que o dongle permaneça conectado enquanto o aplicativo estiver em execução. Você pode definir o intervalo (em segundos) com que a verificação é realizada.

Proteção com pen drive USB genérico

Você pode vincular seu aplicativo a um ou mais pen drives USB genéricos com base no identificador único de fabricante deles.

- Primeiro, você deve inserir um **Application Secret ID**. Esse é combinado com o identificador do pen drive USB para gerar um hash único.
- Quando o aplicativo é iniciado, ele examina todas as unidades USB. Se for encontrada uma unidade com um hash autorizado, a planilha é aberta.

Para autorizar um pen drive USB, conecte-o, selecione-o na lista de unidades do XLS Padlock e clique em "Allow this USB disk". Você pode autorizar vários pen drives USB.

OBSERVAÇÃO

A proteção com pen drive USB oferece menos flexibilidade e segurança do que a proteção com dongle dedicado.

Desativação

Introdução à desativação

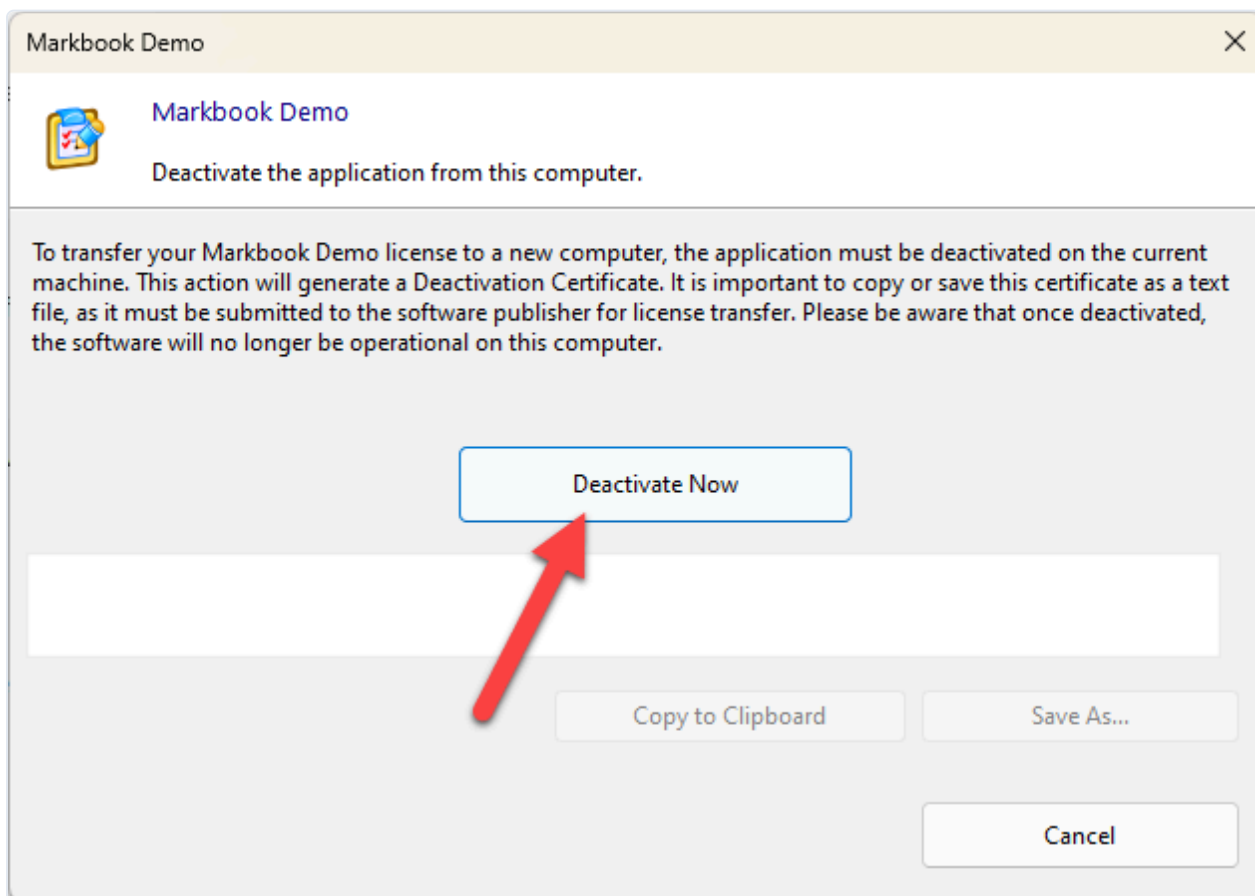
O XLS Padlock oferece um sistema de desativação robusto que permite aos usuários finais cancelar o registro do aplicativo em um computador. Isso é especialmente útil quando os clientes precisam transferir uma licença para uma nova máquina ou reativar o software após mudanças significativas no sistema.

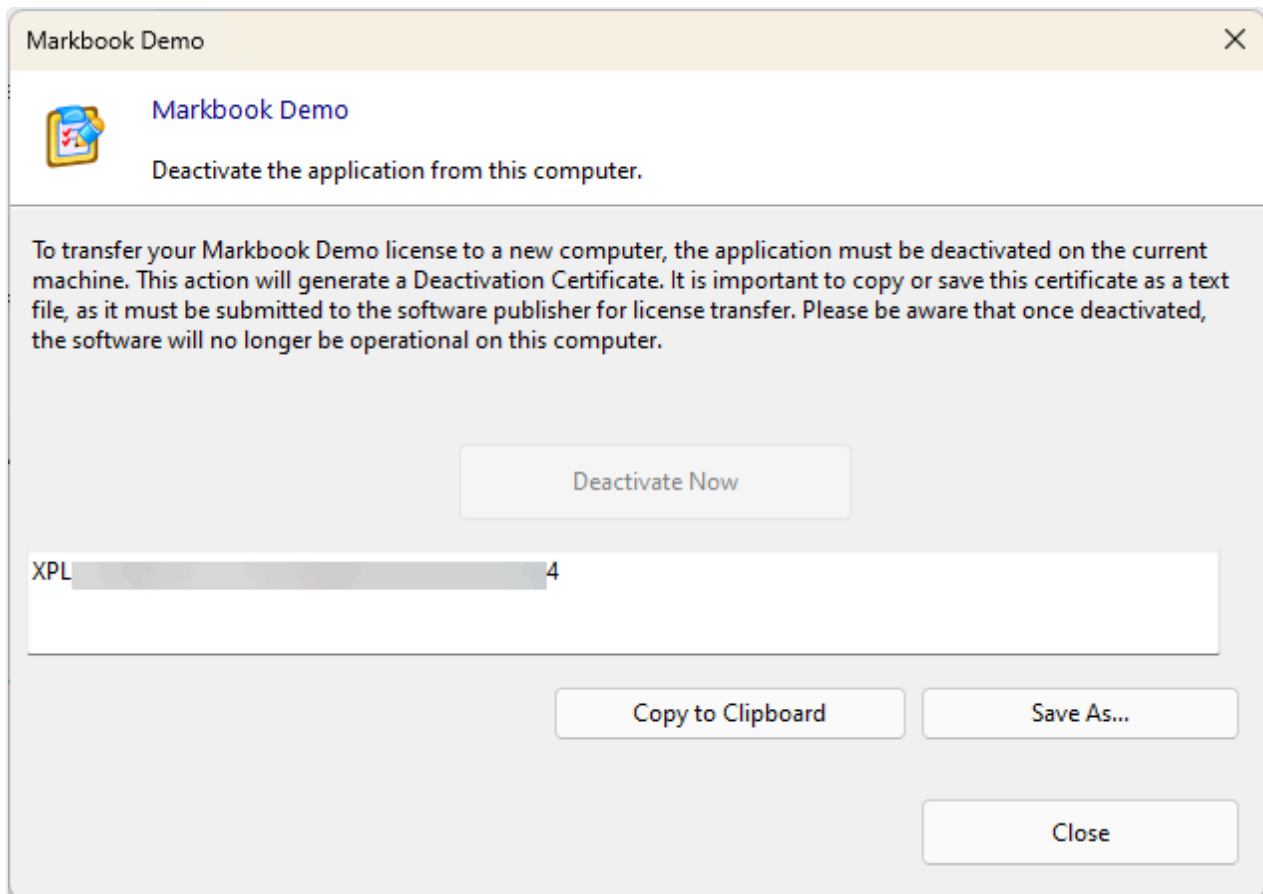
Como funciona a desativação

A desativação gera um certificado que o usuário final deve enviar a você. Por meio do recurso **Test Deactivation Certificate** (testar certificado de desativação) do XLS Padlock, você pode verificar a autenticidade do certificado. É fundamental observar que, uma vez que um aplicativo é desativado, sua chave de ativação é invalidada. Você precisará emitir uma nova chave ao usuário para quaisquer ativações futuras.

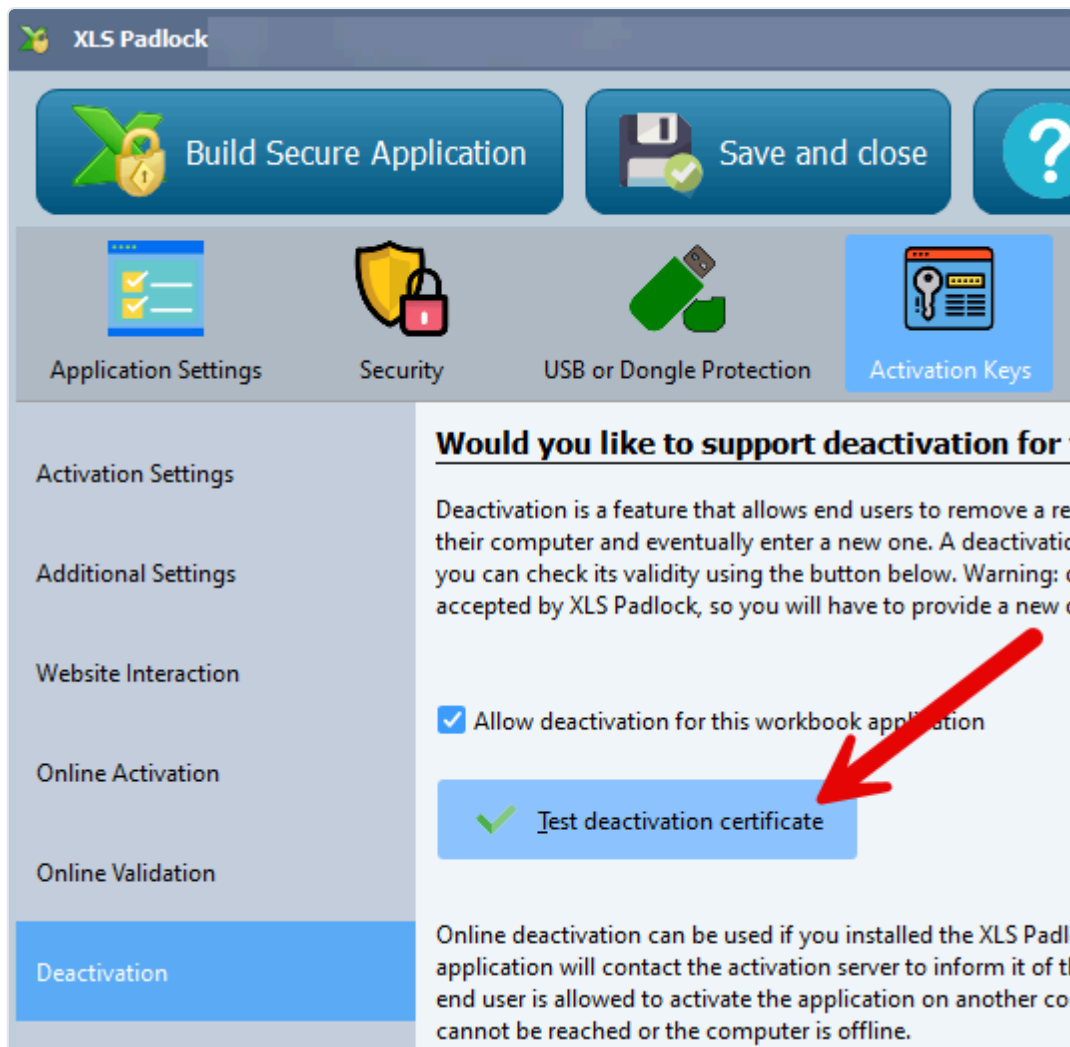
Desativação manual

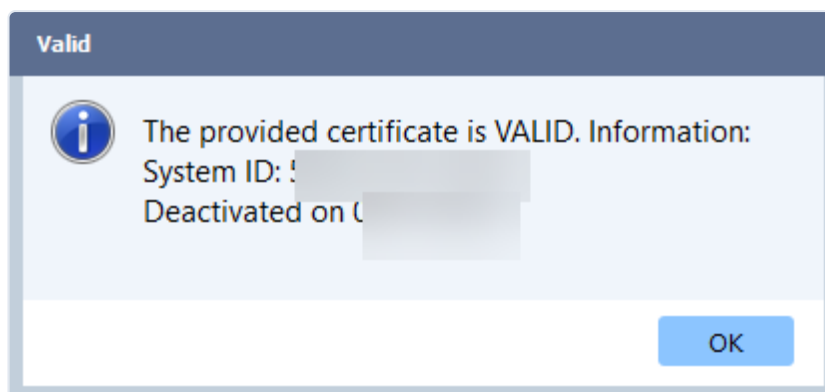
Quando um usuário desativa o aplicativo, ele deve gerar um certificado de desativação e enviá-lo a você:





Ao recebê-lo, você pode usar o botão 'Test deactivation certificate' no XLS Padlock para verificar o certificado. Se a desativação for bem-sucedida, o XLS Padlock exibirá a data de desativação e o identificador único do computador do usuário:

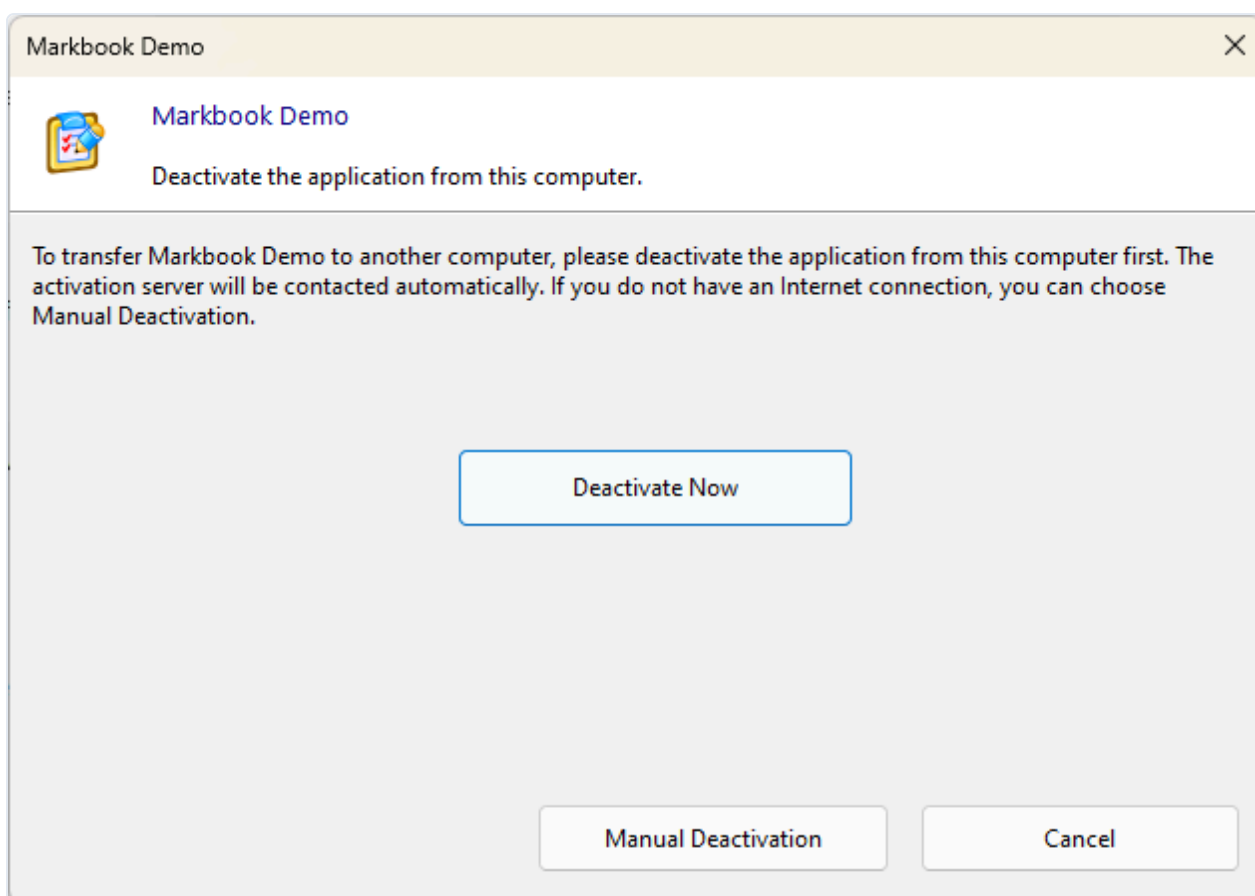




Essas informações permitem que você gerencie novas ativações para o seu cliente.

Desativação on-line

Para uma experiência mais fluida, o XLS Padlock oferece suporte à desativação on-line se você tiver o XLS Padlock Activation Kit instalado no seu servidor web. O aplicativo se comunica com o seu servidor para cancelar automaticamente o registro da ativação. Se o servidor estiver inacessível ou o computador estiver off-line, um certificado de desativação manual é criado como alternativa.



A desativação on-line automatiza todo o processo, facilitando a vida tanto para você quanto para os seus clientes.

Configuração da desativação on-line

Base Deactivation URL

Para usar a desativação on-line, você deve fornecer a URL do XLS Padlock Activation Kit ou do XLS Padlock WooCommerce Integration Kit no seu servidor web. Por exemplo, se você instalou o kit de ativação em uma subpasta chamada "activation", a URL seria

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

Use HTTPS

Há suporte para conexões seguras usando TLS/SSL. Você deve sempre usar URLs que comecem com `https://` .

⚠ Se você preferir gerenciar a desativação manualmente, deixe este campo em branco.

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

Por padrão, a desativação manual está sempre disponível como alternativa. Se você quiser incentivar os usuários a usar a desativação on-line, pode optar por ocultar o botão de desativação manual.

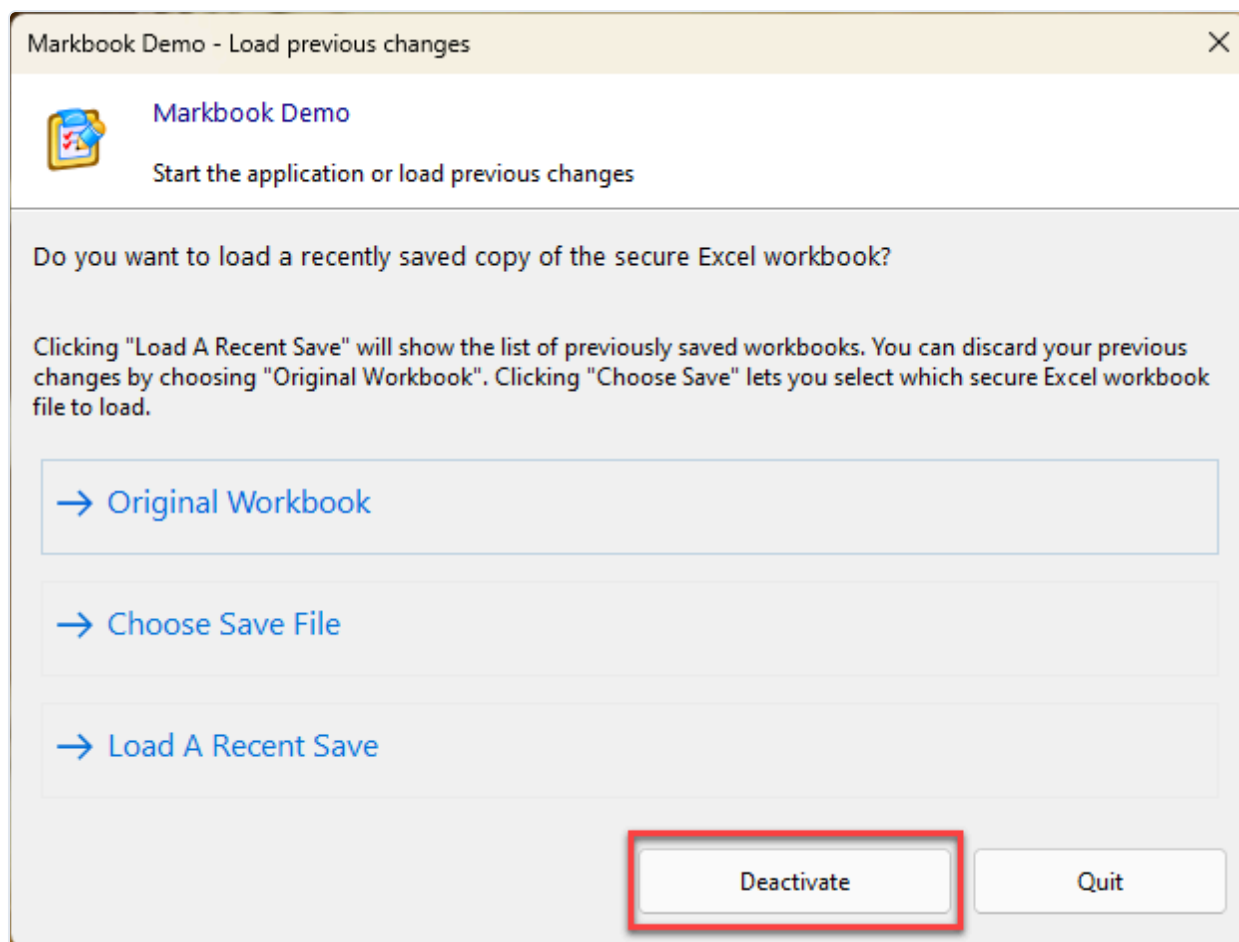
OBSERVAÇÃO

Mesmo que o botão esteja oculto, o aplicativo oferecerá automaticamente a desativação manual se o processo on-line falhar (por exemplo, se o servidor de desativação estiver inacessível ou o usuário estiver off-line).

Como iniciar a desativação

Um usuário pode iniciar o processo de desativação de três maneiras:

- A primeira opção é por meio da caixa de diálogo de boas-vindas integrada do aplicativo. Ao iniciar o aplicativo seguro, a caixa de diálogo de boas-vindas aparece, apresentando várias opções, incluindo um botão "Deactivate". Ao clicar nesse botão, os usuários podem iniciar o processo de desativação de forma simples e guiada:



- O segundo método envolve o uso da linha de comando. Usuários avançados podem preferir esse método para automação ou integração de sistemas. Para isso, o usuário precisa abrir o prompt de comando do Windows, navegar até o diretório onde está localizado o arquivo executável do aplicativo seguro e, em seguida, digitar o nome do executável seguido do parâmetro `** -deact**`. Por exemplo, se o executável se chamar `Application.exe`, o comando completo seria:

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

Ao pressionar Enter, o parâmetro é passado para o arquivo executável, acionando o processo de desativação.

- O terceiro método usa a API de VBA fornecida pelo XLS Padlock.

A desativação é definitiva em uma máquina

Esteja ciente de que, uma vez que uma chave é desativada em um computador, ela é permanentemente incluída em uma lista negra nessa máquina e não pode ser reutilizada. A desativação é uma ação definitiva para esse sistema específico. Para usar o aplicativo no mesmo computador novamente, será necessária uma nova chave de ativação. Observe que isso se refere a chaves de ativação, não a tokens de ativação (conforme definido no WooCommerce Integration Kit).

👉 Veja também:, [Ativação on-line](#chapter-online-activation), A caixa de diálogo de boas-vindas integrada

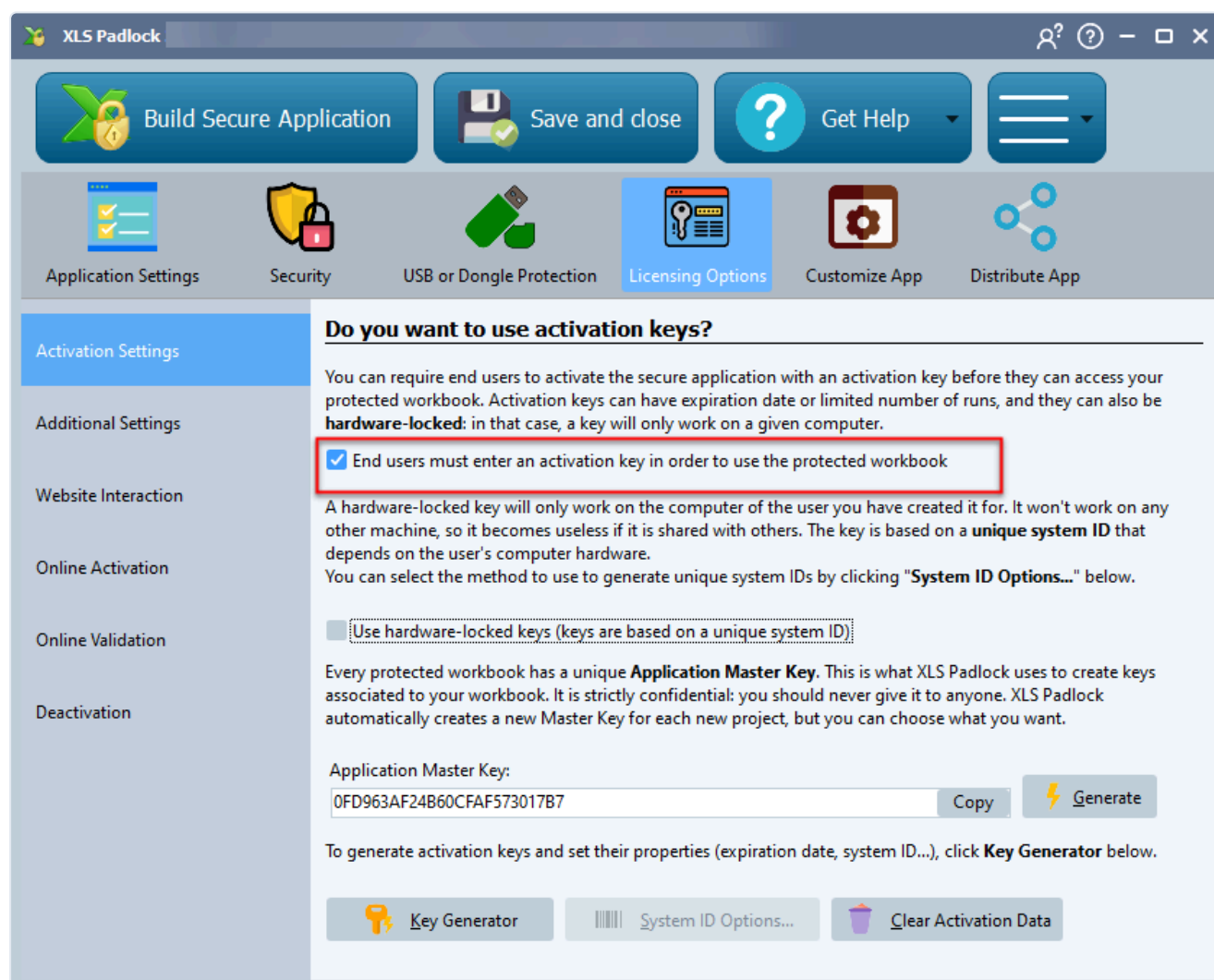
Criar planilhas de avaliação

Você pode oferecer uma prévia das suas planilhas do Excel a clientes potenciais criando uma versão de avaliação que requer uma licença para desbloquear toda a funcionalidade. Isso é possível com as [chaves de ativação do XLS Padlock](#).

Este guia mostra como configurar um período de avaliação para sua aplicação de planilha.

1. Habilitar as chaves de ativação

Primeiro, habilite a opção "**End users must enter an activation key in order to use the protected workbook**" (usuários finais devem inserir uma chave de ativação para usar a planilha protegida) e reconstrua sua aplicação.



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

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

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

A hardware-locked key will only work on the computer of the user you have created it for. It won't work on any other machine, so it becomes useless if it is shared with others. The key is based on a **unique system ID** that depends on the user's computer hardware. You can select the method to use to generate unique system IDs by clicking "**System ID Options...**" below.

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

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

Application Master Key:
0FD963AF24B60CFAF573017B7 Copy ⚡ Generate

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

🔑 Key Generator ⋮ System ID Options... 🗑️ Clear Activation Data

2. Gerar uma chave de ativação de avaliação

Abra o [Key Generator](#) (gerador de chaves) no XLS Padlock.

Para definir um período de avaliação, você pode estabelecer um limite de **Trial days** (dias de avaliação, por exemplo, 15 dias), uma **Expiration Date** (data de expiração) fixa ou um **Max Execution Count** (número máximo de execuções).

XLS Padlock - Application Key Generator

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

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

Max Execution Count: 0

Trial Days: 30

Expiration Date (UTC): 29/11/2020

Display nag screen (useful for trials)

Do not perform online validation for this key

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

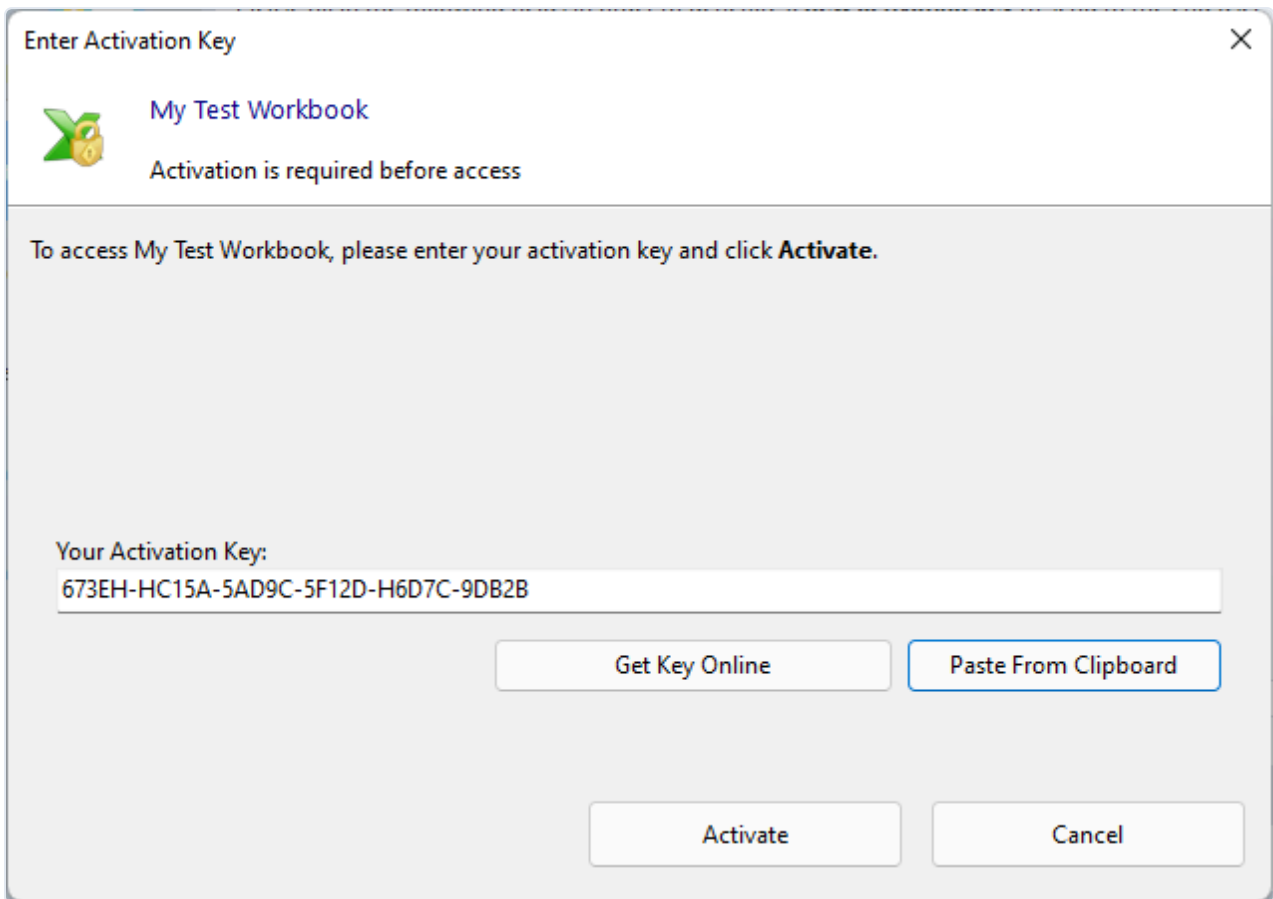
⚡ Generate 📄 Copy to clipboard 💾 Save As... ✅ Close

Para lembrar aos clientes que estão usando uma versão de avaliação, marque **"Display nag screen"** (exibir tela de lembrete). Isso mostrará uma caixa de diálogo de lembrete cada vez que a aplicação for iniciada.

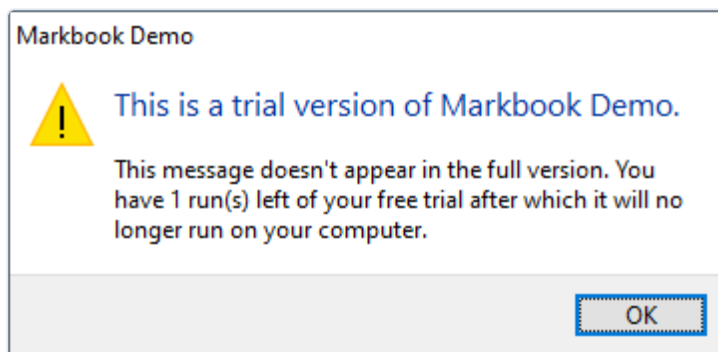
Clique em **Generate** (gerar) para criar a chave de avaliação, que você poderá então distribuir aos seus usuários.

Como a aplicação funciona no modo de avaliação

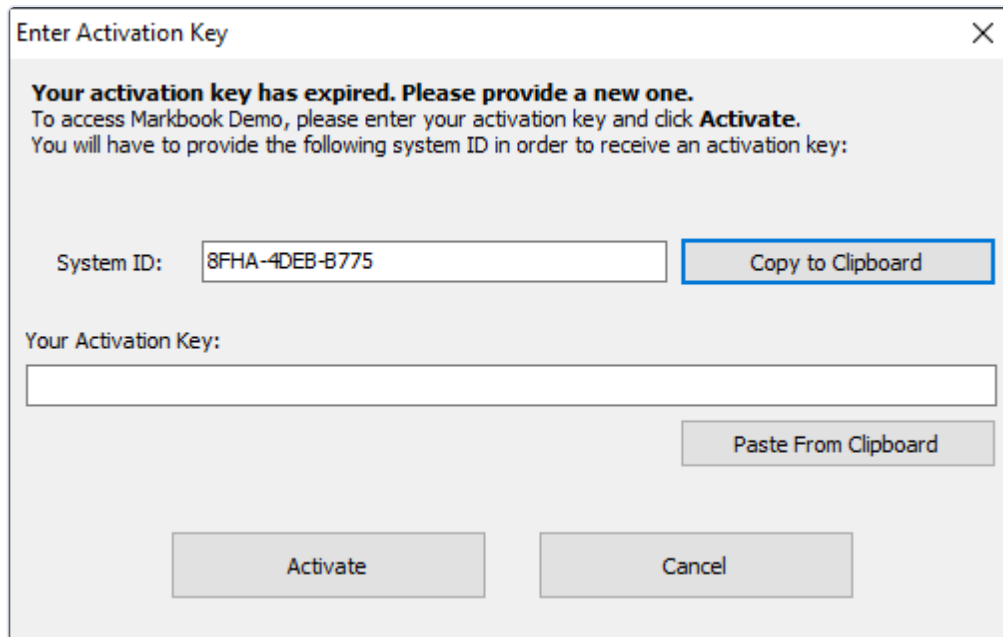
Seus clientes inserirão a chave de avaliação para ativar a aplicação.



Uma vez ativada, a planilha será aberta, mas a tela de lembrete aparecerá para mostrar a duração restante da avaliação.



Quando o período de avaliação terminar, a aplicação exigirá uma nova chave de ativação, que não seja de avaliação, para continuar funcionando.



Enter Activation Key [X]

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

System ID:

Your Activation Key:

Verificar o estado de avaliação com VBA

Você também pode [verificar programaticamente se a planilha compilada está em estado de avaliação](#) usando a API do VBA.

Fechar após um tempo determinado

Como posso impedir que um usuário deixe uma versão de avaliação aberta indefinidamente?

Um usuário poderia obter uma chave de avaliação de 1 dia, abrir o arquivo EXE e deixá-lo aberto para sempre. A aplicação continuaria funcionando. Como isso pode ser evitado?

****Solução**:** Você pode usar VBA para configurar um temporizador com `Application.OnTime` que feche automaticamente a planilha após uma duração determinada. Essa é uma forma eficaz de impor limites de avaliação.

Combine isso com as opções de segurança do XLS Padlock para proibir o acesso ao editor de VBA e compile seu código VBA para impedir que os usuários desativem o temporizador.

Implementação

Insira o código a seguir no módulo `ThisWorkbook` do seu projeto 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
```

Como funciona

1. **Workbook_Open** : Quando a aplicação é iniciada, este evento agenda a macro `CloseAndSave` para ser executada após 8 horas (`TimeValue("08:00:00")`). Você pode ajustar essa duração de acordo com suas necessidades (por exemplo, `TimeValue("01:00:00")` para uma hora).
2. **CloseAndSave** : Esta é a macro que realiza a ação. Ela salva a planilha e depois a fecha.
3. **Workbook_BeforeClose** : Esta é uma etapa de limpeza fundamental. Se o usuário fechar a planilha manualmente, este código cancela o temporizador pendente, evitando que o Excel tente executar uma macro em uma planilha que não está mais aberta, o que causaria um erro.

Verificar o estado de avaliação

O XLS Padlock fornece uma API de VBA que permite determinar de forma programática se a sua planilha compilada está sendo executada em estado de avaliação. Usando a função abaixo, você pode ativar ou desativar recursos específicos, exibir mensagens de atualização ou criar outras limitações de avaliação personalizadas com base no status da licença do usuário...

👉 Insira a seguinte função em um módulo VBA:

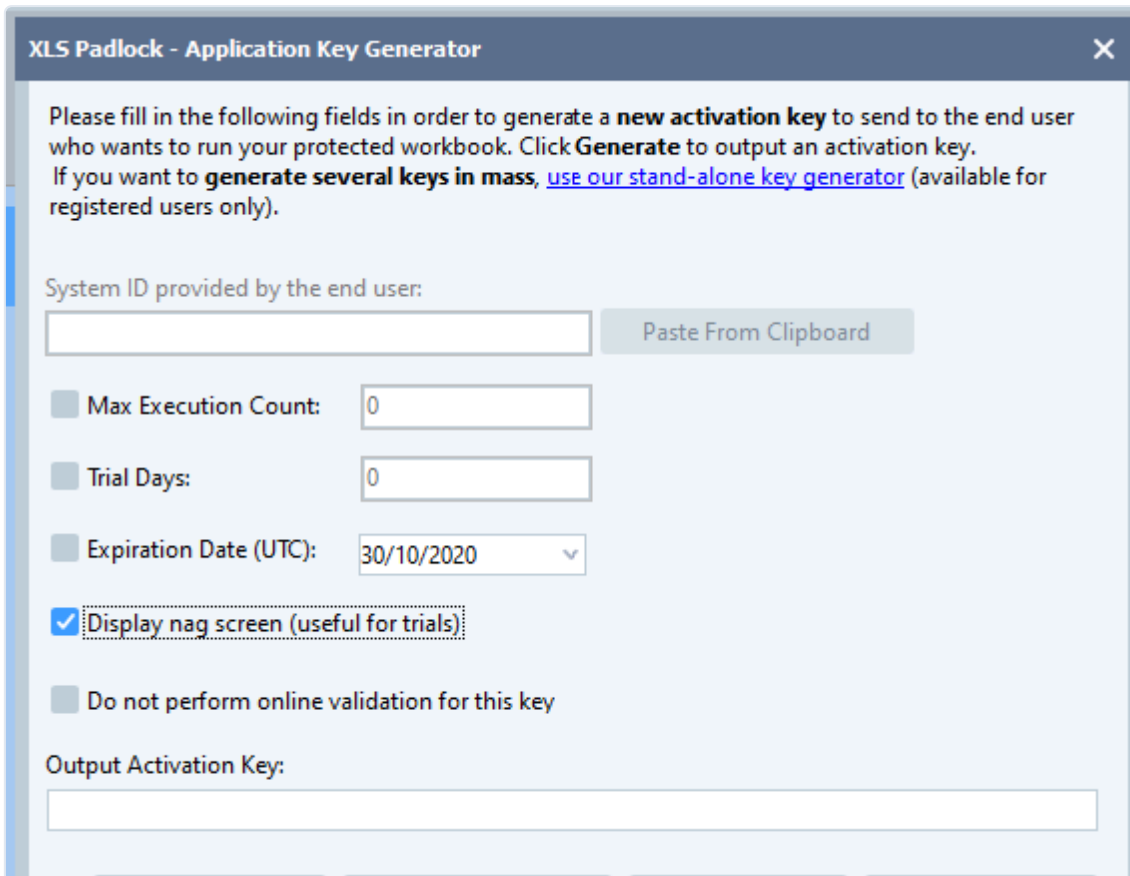
```
Public Function IsTrial()  
    Dim XLSPadlock As Object  
    On Error GoTo Err  
    Set XLSPadlock = Application.COMAddIns("GXLSForm.GXLSFormula").Object  
    IsTrial = XLSPadlock.PLEvalVar("IsTrial")  
    Exit Function  
Err:  
    IsTrial = False  
End Function
```

Em seguida, você pode chamar a função:

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

Aviso

A função retornará true somente se a chave de ativação tiver o indicador "display nag screen" (avaliação) habilitado:



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:

👉 Veja também: [Definir restrições nas chaves de ativação](#)

Dias de avaliação restantes

Esta função VBA permite que você **obtenha de forma programática o número de dias ou execuções restantes** de uma [planilha em versão de avaliação](#). Ela também funciona com chaves registradas que têm uma data de expiração ou um número limitado de execuções.

👉 Insira a função a seguir em um módulo 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
```

Em seguida, você pode chamar a função:

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

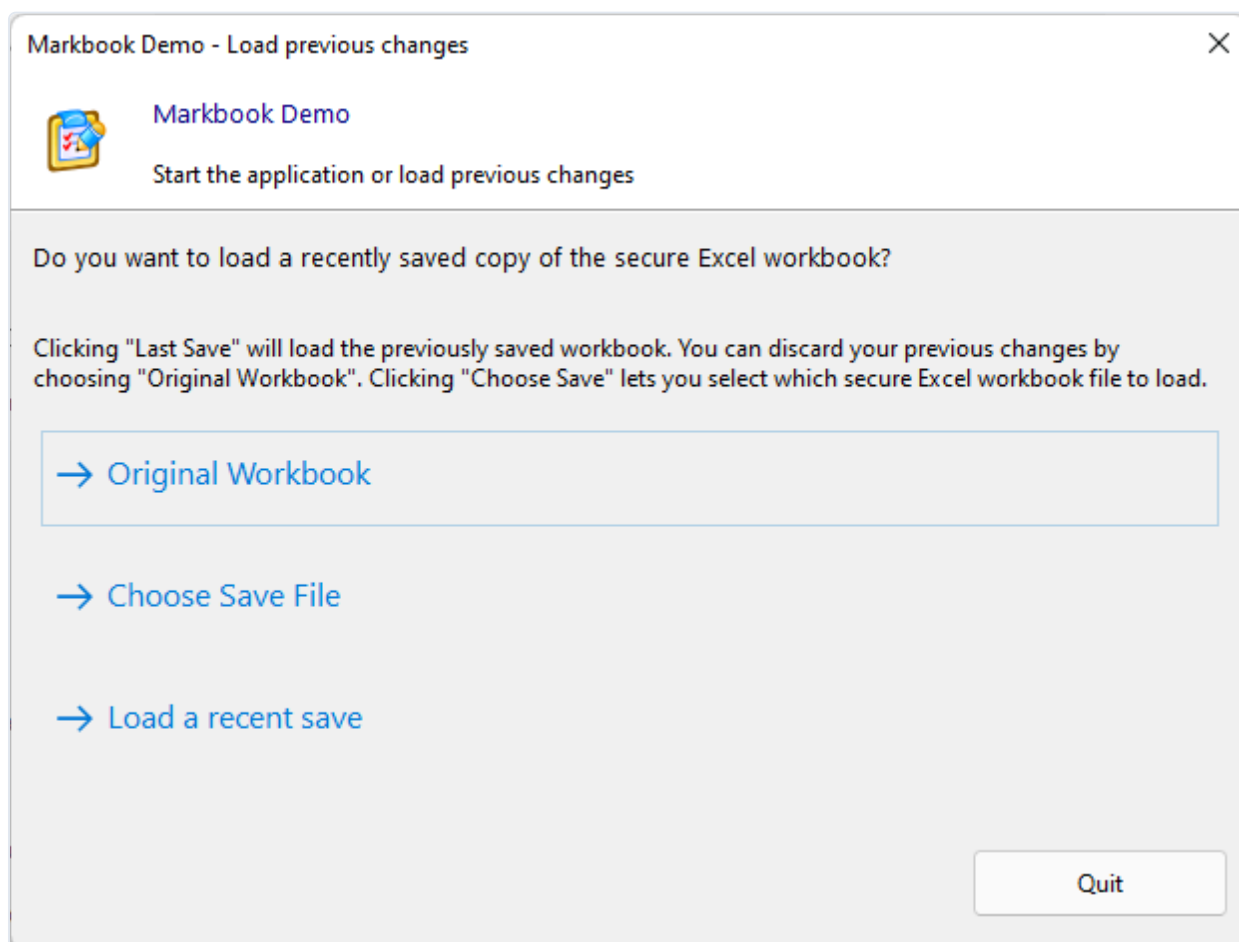
👉 Veja também: [Como criar planilhas em versão de avaliação](#)

Opções de salvamento da planilha

Estas opções permitem que você ajuste com precisão como as modificações da planilha são salvas e carregadas quando você [habilita o salvamento](#).

Always Show the "Load Previous Changes" Welcome Screen

Por padrão, se não existir nenhum arquivo de salvamento, o aplicativo abre diretamente a planilha original. Quando esta opção está habilitada, a caixa de diálogo de boas-vindas é sempre exibida primeiro, permitindo que os usuários comecem com a planilha original ou carreguem um arquivo de salvamento existente.



Isso é útil se você distribui arquivos de salvamento predefinidos com seu aplicativo ou se move o EXE para outro computador e deseja carregar um arquivo de salvamento existente.

Save Changes Automatically and Load Them Without Prompt

Esta opção oferece um mecanismo de salvamento automático e contínuo. As alterações do usuário são salvas em um arquivo padrão quando o aplicativo é fechado e são recarregadas automaticamente na próxima vez que ele é iniciado. Nenhuma caixa de diálogo "Save As" nem tela de boas-vindas é exibida.

Com este modo, os usuários não podem criar vários arquivos de salvamento diferentes.

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

Esta opção aumenta a segurança ao bloquear os arquivos de salvamento no computador específico onde foram criados. Um arquivo de salvamento criado em um PC não pode ser aberto em outro, o que impede que os usuários compartilhem seus dados salvos.

→ Saiba mais sobre o bloqueio por hardware de arquivos de salvamento.

Allow Save but Do Not Handle Loading/Saving

Habilite esta opção se desejar desativar as caixas de diálogo de salvamento integradas do XLS Padlock e gerenciar você mesmo todo o processo de salvamento e carregamento usando código VBA. Isso lhe dá controle total, como salvar uma cópia segura sem nenhuma solicitação ao usuário.

→ Veja como salvar uma cópia segura da planilha sem solicitação usando VBA.

Do Not Display the "Original Workbook" Choice

Esta opção remove a escolha "Start with the original workbook" da tela de boas-vindas, forçando os usuários a carregar um arquivo `.XLSC` ou `.XLSCE` salvo. Isso é útil se o seu aplicativo for sempre destinado a ser usado com arquivos de dados específicos que você fornece.

Rede de segurança automática

Se o arquivo de salvamento anterior de um usuário não for carregado, por exemplo, depois que você alterou a [Secret Key](#) para uma nova versão ou trocou os modos de salvamento entre versões, o botão "Original Workbook" é reabilitado automaticamente na tela de boas-vindas para essa nova tentativa. Isso garante que o usuário sempre possa recuperar o aplicativo em vez de ficar preso em um loop de mensagens de erro.

👉 Veja também

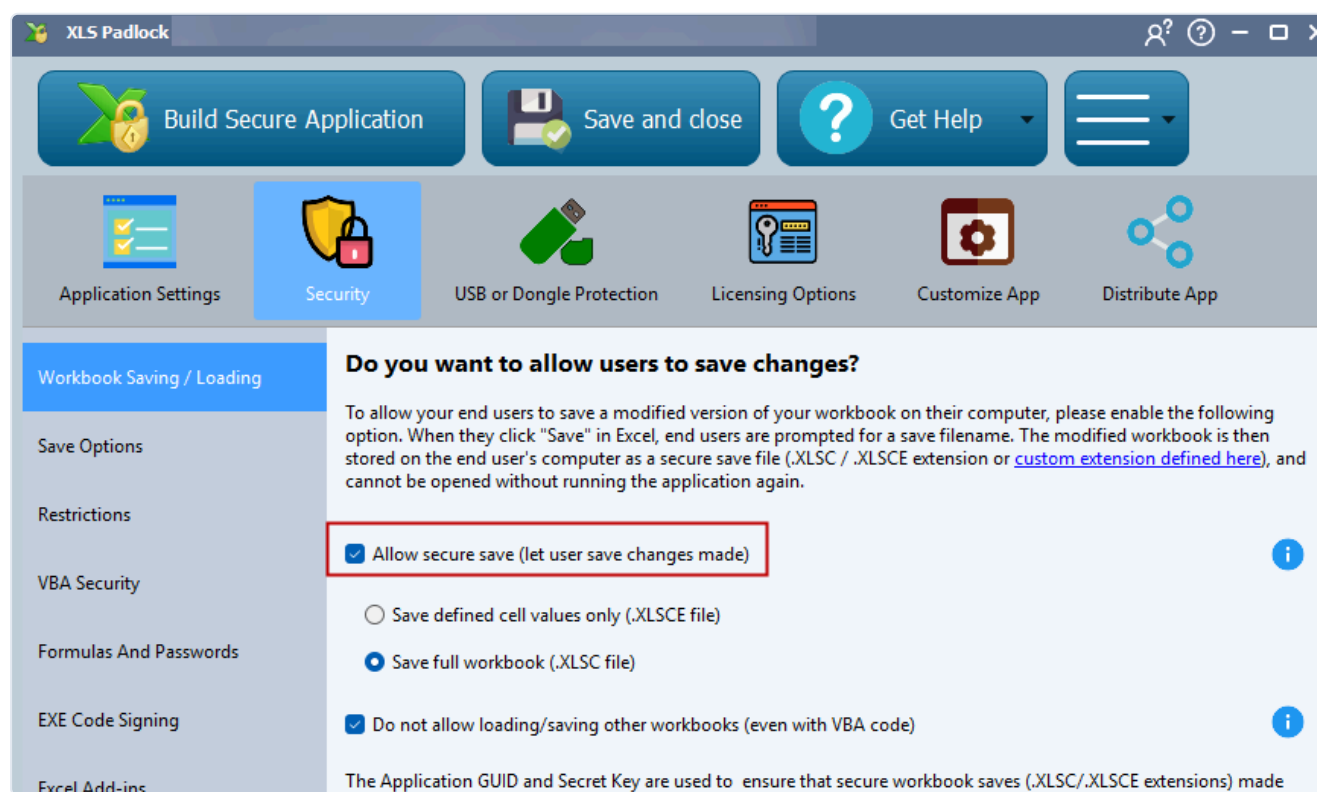
- [Visão geral do salvamento e carregamento de planilhas](#)
- [Modo de salvamento: planilha completa versus valores de célula](#)
- [Receitas VBA para salvar planilhas](#)

Salvar e carregar

Com o XLS Padlock, você tem controle total sobre se os usuários podem salvar alterações na sua planilha do Excel.

Como o XLS Padlock salva e carrega as alterações do usuário

Para habilitar ou desabilitar o salvamento, marque a opção "**Allow secure save**" (permitir salvamento seguro) na página Security:

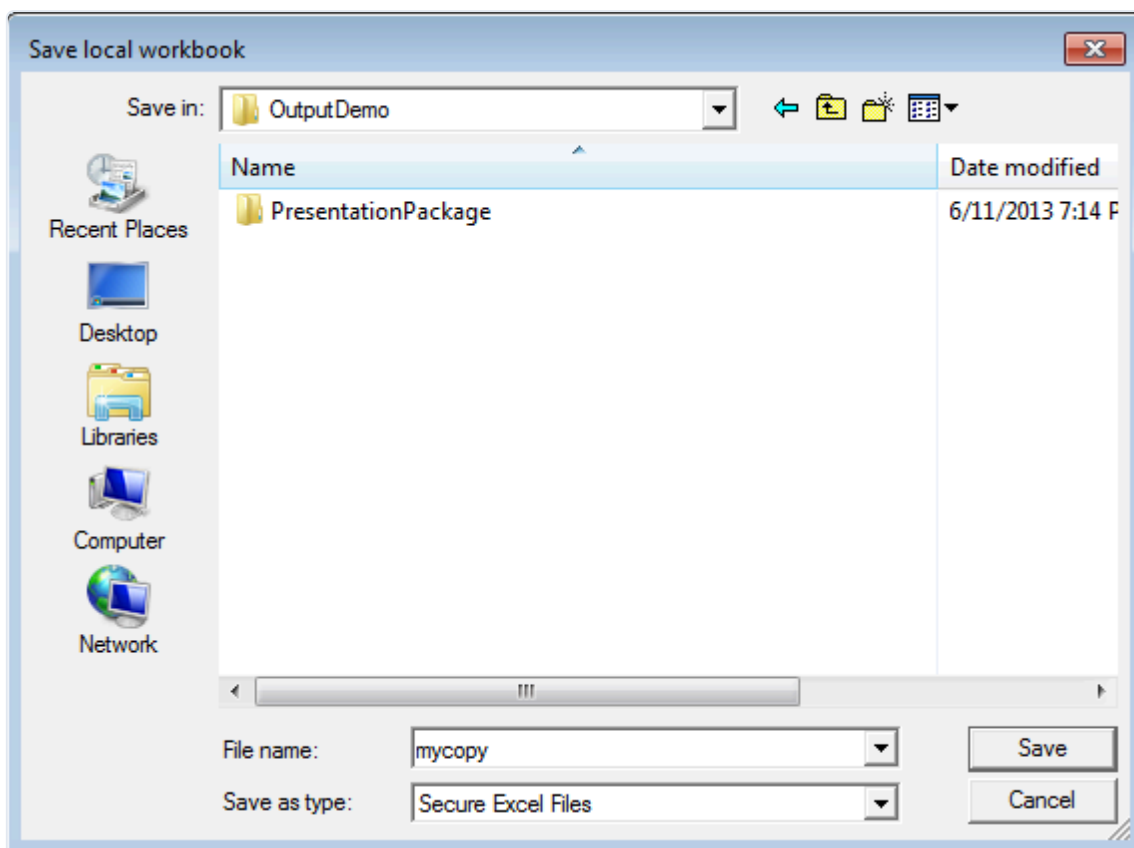


- **Se esta opção estiver desabilitada**, seu aplicativo nunca será modificado. Quaisquer alterações que um usuário faça são descartadas ao fechar o aplicativo. Observe que, em algumas versões do Excel, o botão Salvar não é desativado e parecerá funcionar, mas na verdade nenhuma alteração é salva.
- **Se esta opção estiver habilitada**, seus clientes poderão salvar suas modificações. Em seguida, você precisará [selecionar um modo de salvamento](#).

Para salvar as alterações, os usuários finais podem clicar no botão Salvar padrão



do Excel ou usar o menu "File => Save". A caixa de diálogo "Save As" será então exibida, perguntando onde salvar a cópia segura da planilha:



A este arquivo de salvamento é atribuída a extensão `.XLSC` (ou `.XLSCE`) e ele **não pode ser aberto sem executar o aplicativo seguro novamente**.

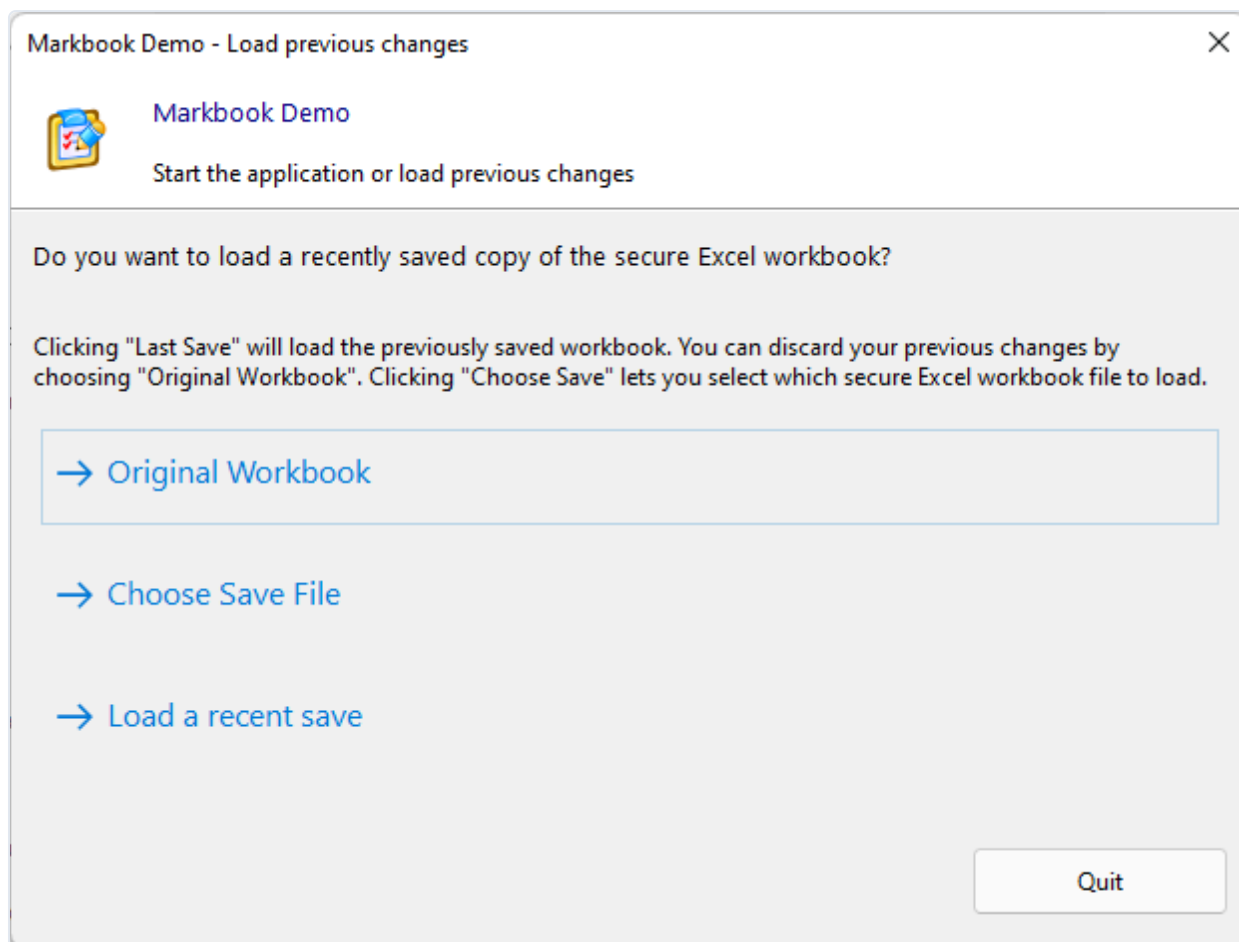
Por padrão, a caixa de diálogo perguntará aos usuários finais se eles desejam sobrescrever suas alterações. Você pode desabilitar esse comportamento nas [Opções avançadas](#).

Extensão personalizada de arquivo de salvamento

Você pode definir sua própria extensão personalizada para arquivos de salvamento nas [Opções avançadas](#).

Como os usuários finais carregam suas alterações

Na próxima vez que executarem seu aplicativo, os clientes poderão escolher entre abrir sua planilha original ou uma versão que salvaram anteriormente. A tela de boas-vindas permite que eles selecionem "Choose Save File" ou carreguem um arquivo a partir do menu "Recent Saves".



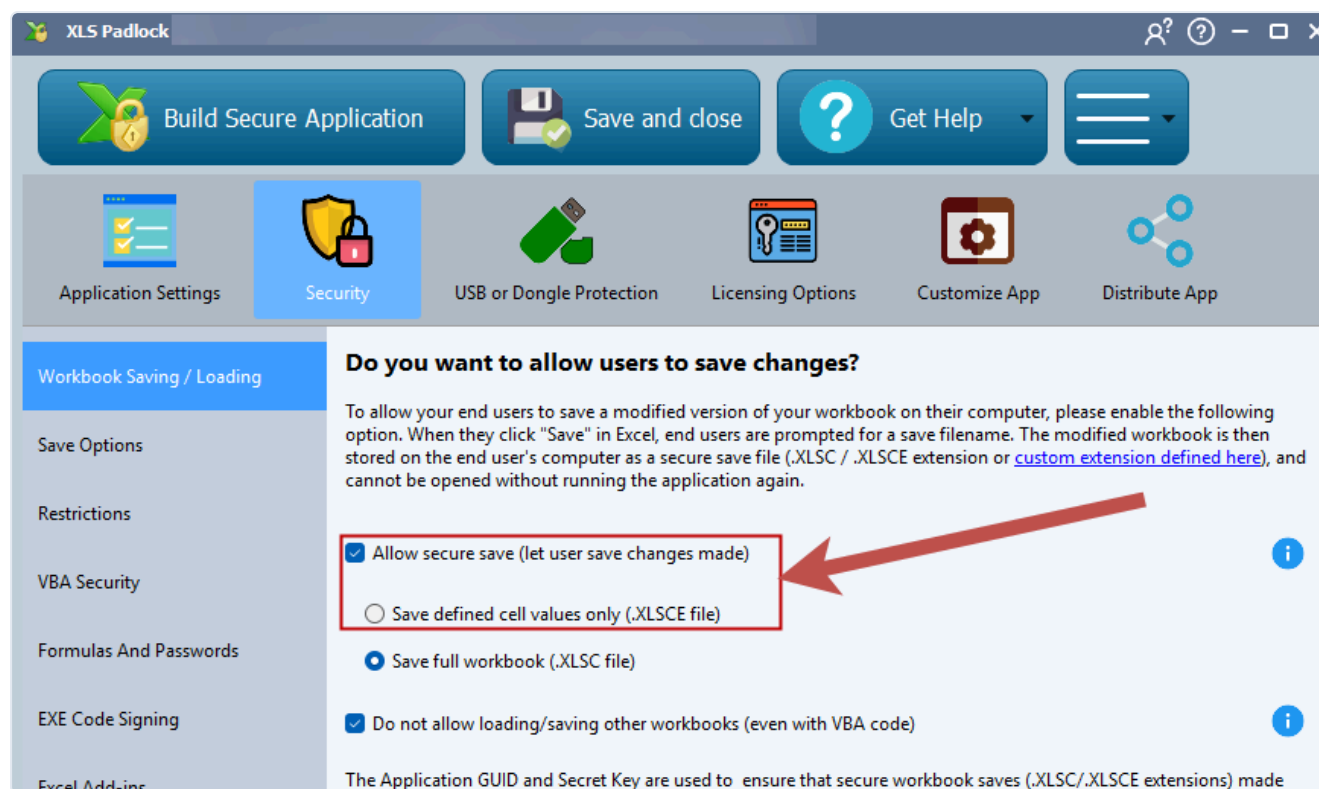
Por padrão, a caixa de diálogo "Save As" é exibida sempre que um usuário clica no botão Salvar. O último arquivo salvo é lembrado automaticamente. Observe que apenas o comando `Save` funciona; `Save As` está sempre desabilitado.

Veja também

- [Como migrar dados do usuário de uma versão anterior com VBA](#)
- [Escolha como os dados são salvos na planilha](#)
- [Salvar e restaurar valores personalizados de forma programática com código VBA](#)

Modo de salvamento: completo ou valores de células

Se você permite que os usuários salvem o seu trabalho, é necessário definir como o XLS Padlock vai salvar e carregar essas alterações. O XLS Padlock oferece dois modos de salvamento, que podem ser selecionados na página [Salvamento e carregamento da planilha](#).



Save Full Workbook (arquivo .XLSC)

Neste modo, uma cópia completa e criptografada da planilha, incluindo todas as alterações do usuário, é armazenada em um arquivo seguro `.XLSC`. Este arquivo só pode ser aberto pelo seu aplicativo protegido, garantindo que a planilha permaneça segura.

O **Full Save mode** (modo de salvamento completo) criptografa e salva toda a planilha exatamente como ela está no momento do salvamento. Isso significa que, se você distribuir mais tarde um EXE atualizado, os usuários que abrirem um arquivo de salvamento antigo verão o trabalho que salvaram anteriormente, e não as suas novas atualizações. O arquivo de salvamento deles é um instantâneo completo da planilha naquele momento.

Os arquivos de salvamento completos podem ser compartilhados entre usuários, a menos que você os [bloqueie a um computador específico](#). Você também pode [criptografar você mesmo esses arquivos de salvamento](#) para recuperar os dados do usuário.

Quando usar o modo Full Save

O modo Full Save é a opção padrão para a maioria dos usuários e é especialmente recomendado para planilhas complexas em que os usuários fazem alterações extensas em várias folhas.

A principal desvantagem é que os arquivos de salvamento do usuário não são automaticamente compatíveis com as versões atualizadas do seu aplicativo. Se você publica atualizações da lógica ou do design da sua planilha com frequência, considere usar o modo Cell Values em vez disso.

👉 Veja também: [Como migrar os dados do usuário de uma versão anterior com VBA](#)

Save Defined Cell Values Only (arquivo .XLSCE)

Neste modo, somente os valores de células específicas e predefinidas são salvos e restaurados. Esta é a escolha ideal se você atualiza com frequência a sua planilha de origem e distribui novas versões.

Quando um usuário carrega o seu arquivo de salvamento `.XLSCE` com o seu novo EXE, os dados salvos são carregados na estrutura atualizada da sua planilha. Isso permite que ele se beneficie das suas atualizações sem perder o seu trabalho.

A principal desvantagem é que você deve [definir manualmente quais células salvar](#) antes de compilar o seu aplicativo.

Assim como os arquivos `.XLSC`, os arquivos `.XLSCE` são criptografados de forma segura e só podem ser abertos pelo seu aplicativo. Isso é imposto por uma [chave secreta única](#) específica do seu projeto.

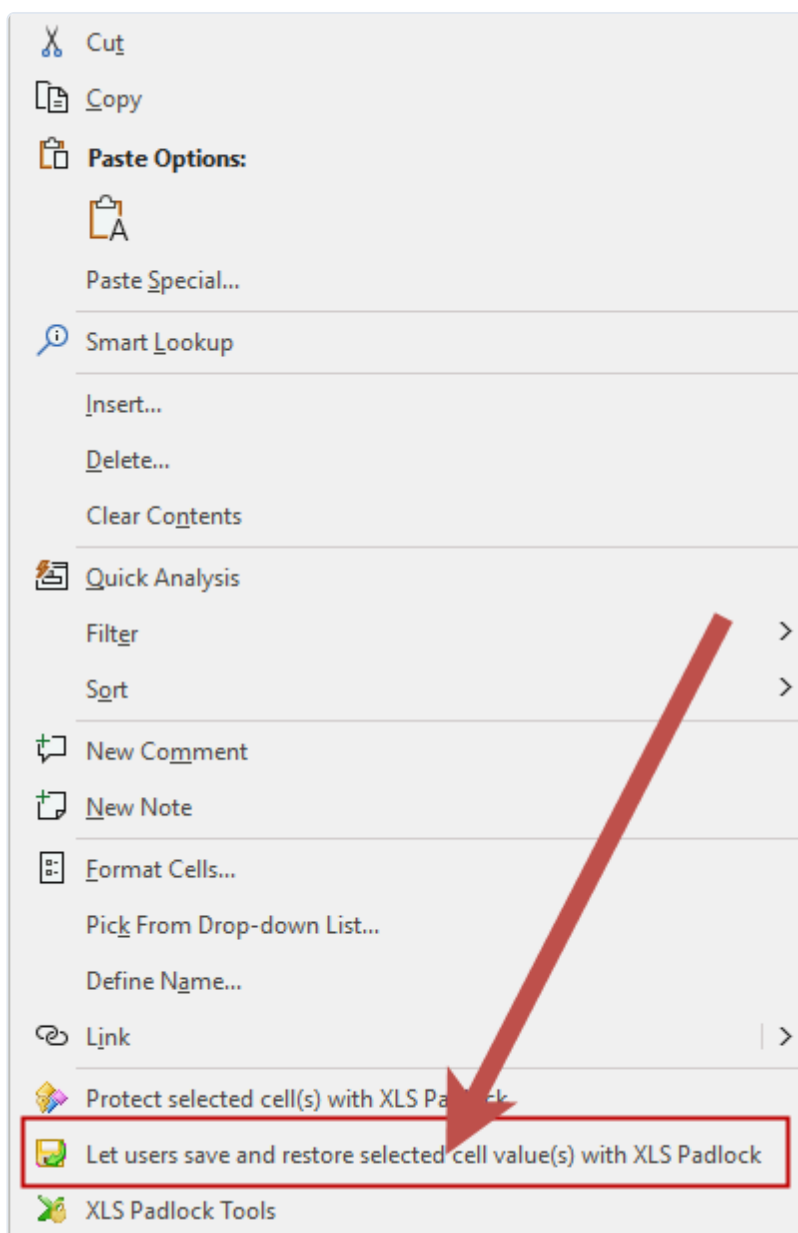
Escolha o seu modo de salvamento de antemão

Os formatos Full Save (`.XLSC`) e Cell Values (`.XLSCE`) não são intercambiáveis. Depois que você tiver distribuído um aplicativo usando um modo, mudar para o outro em uma versão posterior significa que os arquivos de salvamento dos usuários existentes não serão carregados, o XLS Padlock recorrerá silenciosamente à planilha original incorporada para esses usuários na primeira execução da nova versão, e quaisquer dados que eles tenham salvado no modo anterior serão perdidos. Escolha no início do projeto o modo que se ajusta à sua estratégia de atualização e mantenha-o.

Definir células a salvar e restaurar

Ao usar o modo **Save defined cell values only** (salvar apenas valores de células definidas), você deve especificar quais células sua aplicação salvará e restaurará.

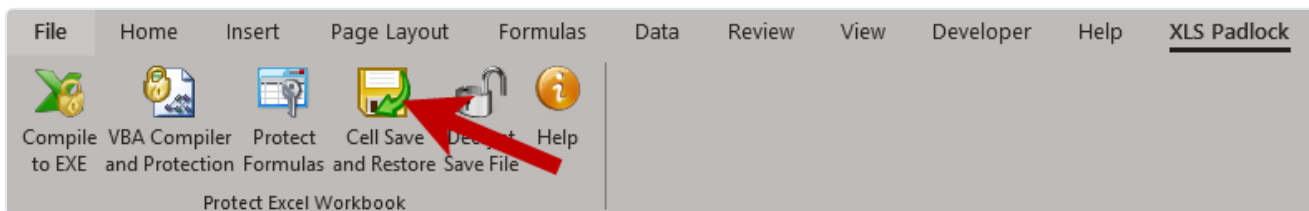
👉 Para isso, basta **clique com o botão direito em uma ou mais células** e escolher **"Let users save and restore selected cell value(s) with XLS Padlock"** (permitir que os usuários salvem e restaurem os valores das células selecionadas com o XLS Padlock) no menu de contexto.



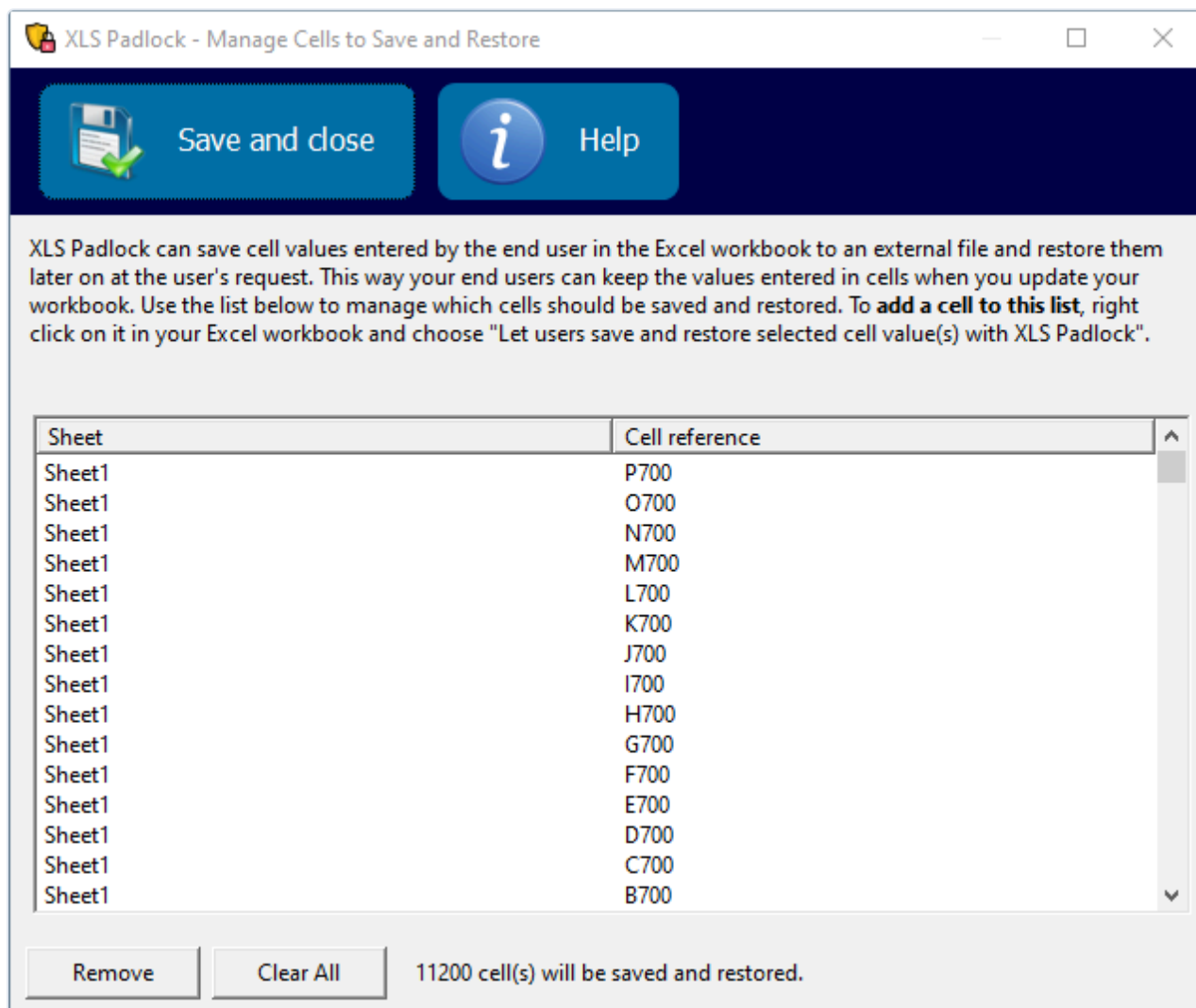
O XLS Padlock confirmará que as células selecionadas estão marcadas para serem salvas e restauradas. Você também pode [armazenar e restaurar valores personalizados com código VBA](#).

Visão geral das células que serão salvas

Para ver todas as células configuradas para serem salvas, clique em "Cell Save And Restore" (salvar e restaurar células) na aba da faixa de opções do XLS Padlock:



Isso abre uma janela que lista todas as células configuradas. A partir daqui, você pode remover células individuais ou limpar a lista inteira.



Restaurar e salvar com VBA

Ao usar o modo **Save defined cell values only** (salvar apenas valores de célula definidos), o XLS Padlock permite que você salve e restaure valores personalizados, além dos valores de célula predefinidos. Isso é útil para salvar variáveis, configurações ou dados que não são armazenados diretamente em uma célula.

Visão geral

Essa funcionalidade depende de dois eventos VBA e de duas funções da API VBA. Você coloca o código nas sub-rotinas de evento para ler ou gravar seus dados personalizados quando o usuário carrega ou salva o trabalho.

Eventos e funções principais

Eventos VBA

As duas sub-rotinas de evento a seguir devem ser colocadas em um módulo da sua planilha do Excel. O XLS Padlock vai chamá-las automaticamente durante o processo de salvar/carregar.

```
' 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
```

Funções da API VBA

- **WriteCustomCellValue(UniqueID, Value)**: grava um único valor de cadeia (string) associado a um ID exclusivo.
- **ReadCustomCellValue(UniqueID, DefaultValue)**: lê um único valor de cadeia para um determinado ID. Se o ID não for encontrado, retorna o `DefaultValue`.

Importante

`WriteCustomCellValue` e `ReadCustomCellValue` só funcionam quando chamadas de dentro dos eventos `XLSPadlock_SaveCustomValues` e `XLSPadlock_RestoreCustomValues`, respectivamente.

Leitura e gravação de valores individuais

Veja como gravar um único valor:

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

E veja como lê-lo de volta:

```
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
```

Leitura de todos os valores como um dicionário

Você também pode ler todos os valores personalizados salvos de uma vez, passando uma cadeia vazia como ID para `ReadCustomCellValue`. Isso retorna um objeto `Scripting.Dictionary`.

Referência necessária

Para usar esse recurso, você deve adicionar uma referência ao **“Microsoft Scripting Runtime”** no seu projeto VBA (no menu do VBE, vá em 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

```
## Exemplo completo: salvar/restaurar uma coluna
```

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

Todo o código a seguir deve ser colocado em um único módulo.

```
</div>
```

Esta função auxiliar gera uma cadeia separada por vírgulas a partir dos valores de um determinado

```
`` `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 é chamado quando o usuário salva o trabalho. Ele usa a função auxiliar para converter todo o intervalo usado da Coluna A em uma única cadeia e a salva.

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

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

    Dim myString As String
    myString = CsvRange(rng)

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

Este evento é chamado quando o usuário carrega um arquivo salvo. Ele lê a cadeia e restaura os valores de volta na Coluna 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 também

- [Como migrar dados de usuário de uma versão anterior](#chapter-excel-vba-migrate-user-data-up)

Acessar a planilha segura e os arquivos complementares

OBSERVAÇÃO

Este guia explica como iniciar uma nova instância do Excel a partir da sua planilha protegida e conceder a ela acesso à planilha segura e aos seus arquivos complementares.

Por padrão, somente a instância do Excel iniciada pelo seu EXE compilado pode acessar a planilha segura e seus arquivos associados. Se você criar uma nova instância do Excel por meio do VBA, ela não terá permissão para abrir esses arquivos protegidos.

Para conceder o acesso, você deve usar a API do XLS Padlock para passar o identificador de janela (HWND) da nova instância do Excel para o aplicativo principal. Isso autoriza a nova instância a trabalhar com o sistema de arquivos 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

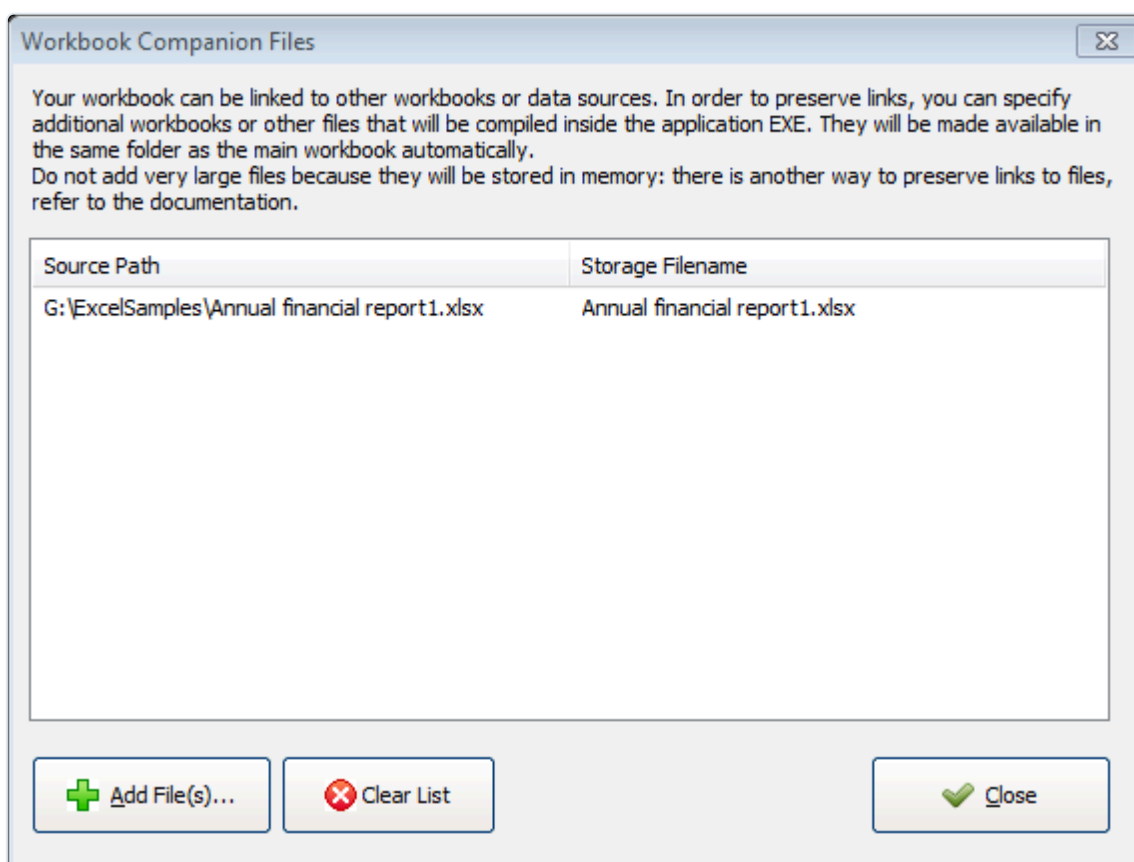
Public Function GetPathToFileInEXEFolder(ByVal Filename As String) As String
' Helper function to get the full path to a file in the EXE's directory.
' See: https://www.xlspadlock.com/doc/get-the-path-to-a-file-in-the-same-folder-as-the-comp
Dim XLSPadlock As Object
On Error Resume Next
Set XLSPadlock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

Adicionar arquivos complementares

Se sua planilha está vinculada a outras planilhas ou fontes de dados, você pode adicioná-las como "arquivos complementares". Isso as integra diretamente no EXE do aplicativo principal, facilitando a distribuição. Quando o aplicativo é executado, esses arquivos ficam disponíveis na mesma pasta virtual que a planilha principal.

Você pode até incluir suplementos XLL.

Por exemplo, se sua planilha usa um arquivo de texto externo como fonte de dados, adicioná-lo como arquivo complementar garante que ele sempre será encontrado quando a planilha compilada for aberta em qualquer computador.



Para adicionar arquivos, clique em **Add Files** (adicionar arquivos) e selecione-os. Eles aparecerão na lista. A coluna "Storage Filename" mostra o nome de arquivo que será usado quando o arquivo for compilado no EXE e restaurado em tempo de execução.

Evite arquivos grandes

Não adicione arquivos muito grandes, pois eles serão armazenados inteiramente na memória. Para vincular outras planilhas ou arquivos de dados de grande tamanho, é melhor colocá-los na mesma pasta do EXE do seu aplicativo e referenciá-los usando a função `PLEvalVar("XLSPATH")`. Consulte [Usar referências externas e hiperlinks](#) para obter mais detalhes.

Acessar arquivos complementares com VBA

Para acessar os arquivos complementares com VBA, você pode usar a seguinte função:

```
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
```

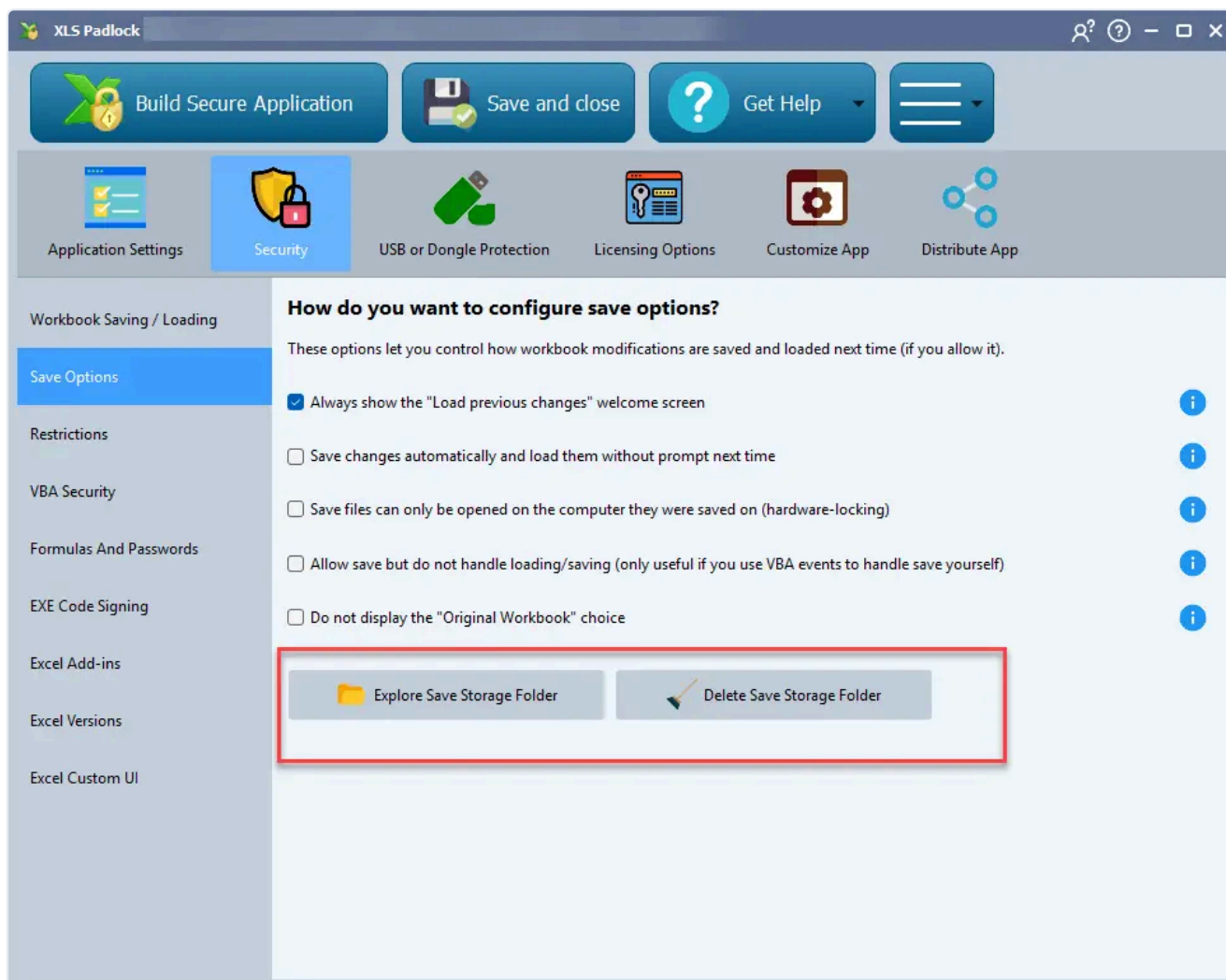
Em seguida, você pode usar essa função para abrir um arquivo complementar:

```
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
```

Pasta de armazenamento de salvamentos

Os botões **Explore Save Storage Folder** (Explorar pasta de armazenamento de salvamento) e **Delete Save Storage Folder** (Excluir pasta de armazenamento de salvamento) fornecem acesso rápido à pasta local no seu PC onde a sua aplicação protegida armazena seus dados.



Esta pasta contém: arquivos salvos automaticamente (se esse recurso estiver habilitado), o arquivo de histórico de salvamento, dados de licença e ativação.

Se o modo portátil estiver habilitado (em [Opções avançadas](#)), esses botões abrirão a mesma pasta que contém o arquivo EXE da aplicação. Por segurança, o botão "Delete" (Excluir) fica desabilitado nesse modo.

Armazenar as alterações dentro do EXE

Não, os usuários não podem salvar alterações diretamente no arquivo `.exe` compilado. Modificar um arquivo executável após a compilação pode corrompê-lo e, com frequência, provoca falsos positivos em softwares antivírus.

Por esse motivo, o XLS Padlock foi projetado para salvar os dados do usuário em arquivos de salvamento criptografados separados (com extensões `.xlsc` ou `.xlsce`), e não de volta no EXE da aplicação principal.

Considere o seu EXE compilado como a própria aplicação (como o `Excel.exe`), e os arquivos de salvamento como os documentos com os quais ela trabalha.

Restringir o carregamento e o salvamento

Uma vulnerabilidade em potencial é que as planilhas do Excel podem ser salvas em disco usando comandos VBA ou OLE. Para impedir esse método de extração de dados, o XLS Padlock inclui uma poderosa opção de segurança chamada **"Do not allow loading/saving other workbooks"** (Não permitir o carregamento/salvamento de outras planilhas).

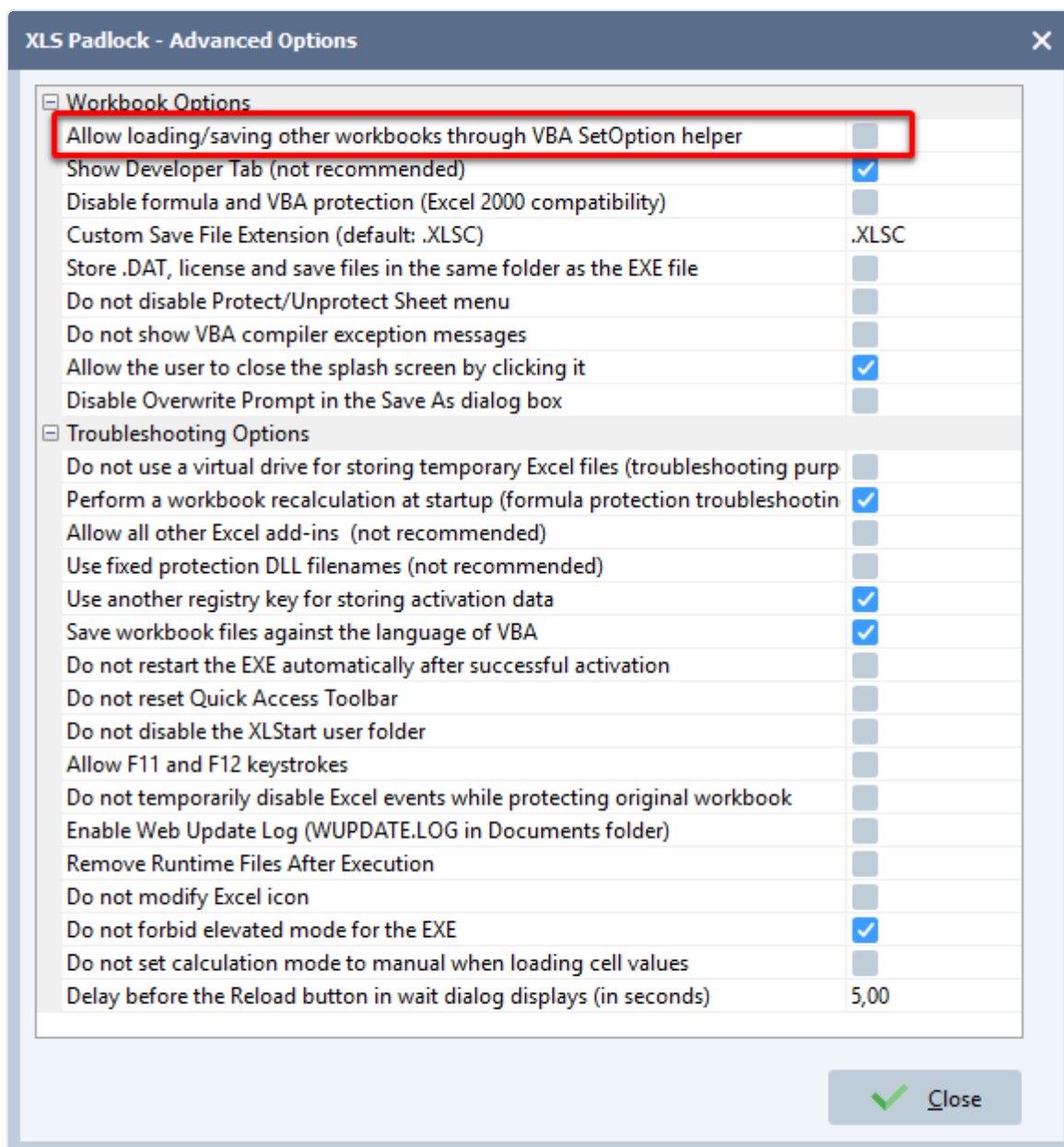
Quando habilitado, esse recurso faz com que a instância do Excel que executa sua aplicação fique **incapaz de carregar ou salvar qualquer outro arquivo de planilha**. Isso bloqueia de forma eficaz que os usuários abram outras planilhas e impede hacks baseados em VBA que tentam copiar dados para um novo arquivo sem proteção.



E se minha aplicação precisar salvar outras planilhas? A solução VBA SetHelper

Esse recurso de segurança também bloqueia o código VBA padrão de salvar ou carregar planilhas. Se sua aplicação precisar legitimamente realizar essas ações, há uma solução alternativa disponível.

Você pode desabilitar temporariamente a restrição dentro do seu código VBA. Para isso, primeiro é necessário habilitar a opção avançada **"Allow loading/saving other workbooks through VBA SetOption helper"** (Permitir o carregamento/salvamento de outras planilhas por meio do helper VBA SetOption) aqui:



Em seguida, você pode usar o seguinte código VBA:

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

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

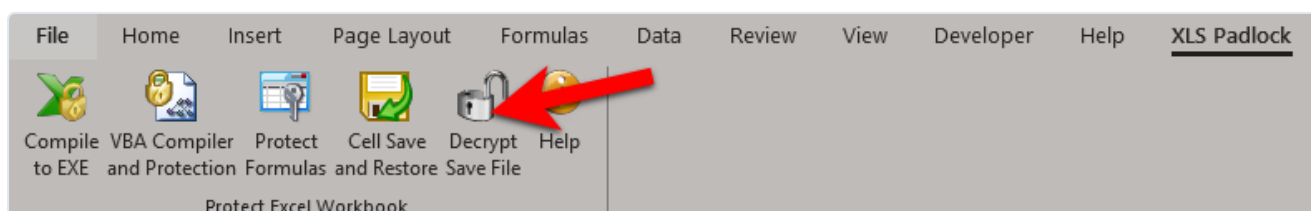
' Your code to save a normal workbook
ActiveWorkbook.SaveAs "D:\My Documents\NormalWorkbook.xlsx"

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

Abrir e descriptografar os salvamentos

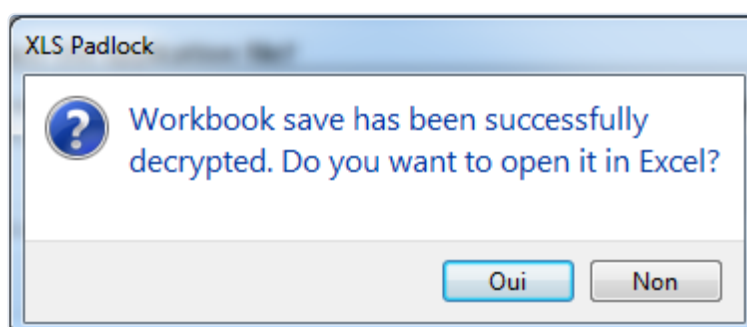
Os arquivos de salvamento criados pelo seu aplicativo são criptografados e não podem ser abertos diretamente no Excel. No entanto, como autor original da planilha, **você pode descriptografar qualquer arquivo de salvamento gerado pelo seu aplicativo**. Isso é útil para recuperar dados caso um cliente lhe envie o arquivo de salvamento dele.

Para abrir um arquivo de salvamento (`.xlsc` ou `.xlsce`), escolha "**Decrypt Save File**" (Descriptografar arquivo de salvamento) na faixa de opções do XLS Padlock no Excel:



Descriptografar arquivos `.XLSC` (modo de salvamento completo)

Se você selecionar um arquivo `.xlsc` criado com o [modo de salvamento completo](#), o XLS Padlock descriptografará a planilha inteira e a abrirá.



Limitações da descriptografia

- As fórmulas protegidas com o XLS Padlock não podem ser recuperadas ao descriptografar um arquivo de salvamento. A planilha descriptografada pode não ser totalmente funcional. Este recurso deve ser usado apenas para recuperar os dados inseridos pelo usuário.
- A mesma limitação se aplica ao projeto VBA caso você o tenha bloqueado ou compilado.
- Para usar a planilha com todos os recursos intactos, você deve executar seu aplicativo seguro e carregar o arquivo de salvamento normalmente.

Descriptografar arquivos `.XLSCE` (modo de valores de célula)

Se você selecionar um arquivo `.xlsce` criado com o modo de salvamento de valores de célula, o XLS Padlock perguntará se você deseja sobrescrever as células da planilha atualmente aberta com os valores do arquivo de salvamento.

Isto sobrescreverá seus dados

Confirmar esta operação sobrescreverá permanentemente os valores de célula na planilha de origem aberta.

Arquivo de projeto necessário

Este recurso de descryptografia requer o arquivo de projeto original do XLS Padlock (.xplp) que foi usado para criar o aplicativo. O XLS Padlock usa as configurações do seu projeto (como a Secret Key) para criptografar e descryptografar os arquivos de salvamento. Sem o arquivo de projeto correto, você não pode descryptografar os salvamentos. **Nunca compartilhe seus arquivos de projeto do XLS Padlock.**

Bloquear os salvamentos a uma máquina

Esta opção impede que os usuários finais compartilhem seus arquivos de salvamento. Quando ativada, o aplicativo cria **arquivos de salvamento bloqueados por hardware**, que só podem ser abertos no computador onde foram criados. Se um usuário tentar abrir um arquivo de salvamento em um computador diferente, ele não será carregado.

O aplicativo incorpora o **ID de sistema** único do computador no arquivo de salvamento e o verifica ao carregá-lo.

DICA

Como proprietário do aplicativo, você ainda pode descriptografar você mesmo qualquer arquivo de salvamento bloqueado por hardware usando o recurso [Decrypt Save File](#) do XLS Padlock.

Aviso

Este recurso só está disponível se você tiver selecionado o modo Full Save (salvamento completo).

Referências externas e hiperlinks

O XLS Padlock protege uma planilha por arquivo EXE. Se a sua planilha utiliza referências externas ou requer arquivos adicionais, você deve atualizar os seus caminhos para garantir que funcionem corretamente após a compilação.

O XLS Padlock oferece duas formas principais de lidar com arquivos externos:

- Adicioná-los como [Companion Files](#).
- Utilizar caminhos dinâmicos em hiperlinks ou código VBA.

Para gerenciar caminhos dinâmicos, o XLS Padlock oferece uma função chamada `PLEvalVar` que pode ser utilizada diretamente em fórmulas do Excel ou chamada a partir do VBA.

Esta função recebe um argumento de cadeia de texto:

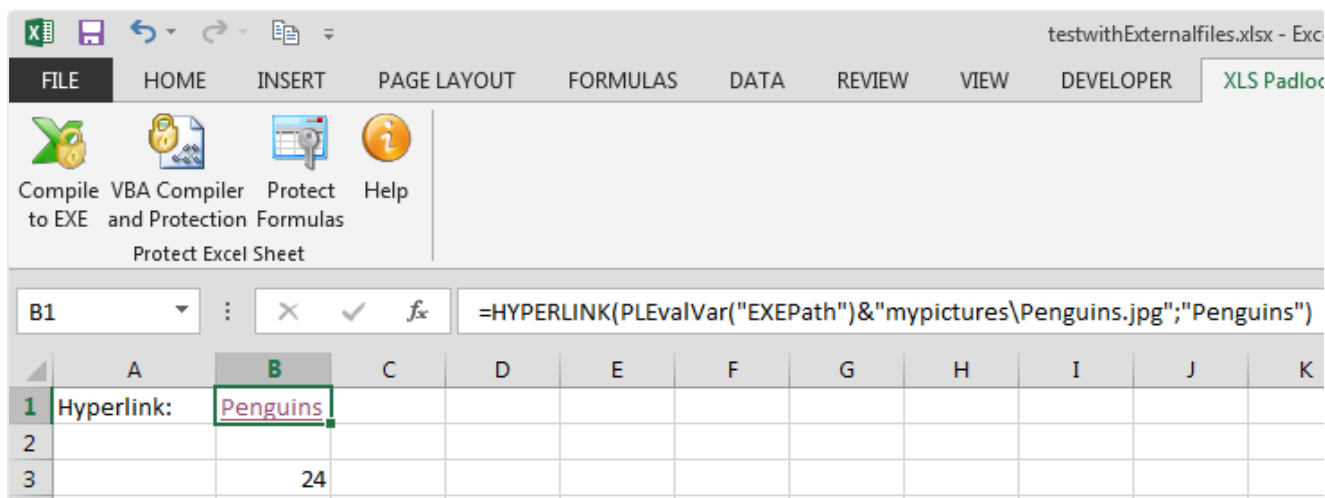
- `=PLEvalVar("EXEPath")` retorna o caminho completo para a pasta que contém o arquivo EXE da aplicação (com a barra invertida final).
- `=PLEvalVar("XLSPath")` retorna o caminho completo para a pasta que contém a planilha compilada em tempo de execução (com a barra invertida final).

OBSERVAÇÃO

Esta pasta é uma pasta virtual e, portanto, você não pode colocar arquivos reais nela. Ela só é útil se você trabalhar com Companion files (consulte [Add Companion Files](#)).

Exemplo 1

Você tem hiperlinks para arquivos de imagem externos. Esses arquivos de imagem estão na mesma pasta que o arquivo XLS da planilha (ou em uma subpasta).



Você tem um hiperlink em uma célula definido por:

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

Para que funcione com o XLS Padlock, você deve copiar todos os arquivos de imagem externos para a mesma pasta que o arquivo EXE gerado com o XLS Padlock. Em seguida, você precisa modificar todos os hiperlinks para inserir a função `PLEvalVar("EXEPath")` que retorna o caminho para essa pasta.

No nosso caso, isso se tornará:

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

Aviso

Os arquivos externos devem ser implantados na mesma pasta que o arquivo EXE final da aplicação. Também é uma boa prática evitar o uso de espaços nos nomes dos arquivos.

Isso também funciona para arquivos em subpastas. Um link como:

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

...deve ser alterado para:

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

Exemplo 2

Para acessar arquivos externos com VBA, você pode utilizar uma função auxiliar que constrói o caminho completo para um arquivo localizado na mesma pasta que o EXE.

👉 Para obter uma explicação detalhada e um trecho de código reutilizável, consulte o guia sobre como obter o caminho para um arquivo na pasta do 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
```

Obter o caminho ao lado da planilha compilada

Esta função VBA recupera o caminho completo para um arquivo localizado no mesmo diretório que o EXE da sua aplicação. Ela é especialmente útil para acessar recursos externos ou arquivos complementares que você distribui junto com a sua planilha protegida.

👉 Insira a seguinte função em um 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
```

Em seguida, você pode chamar a função:

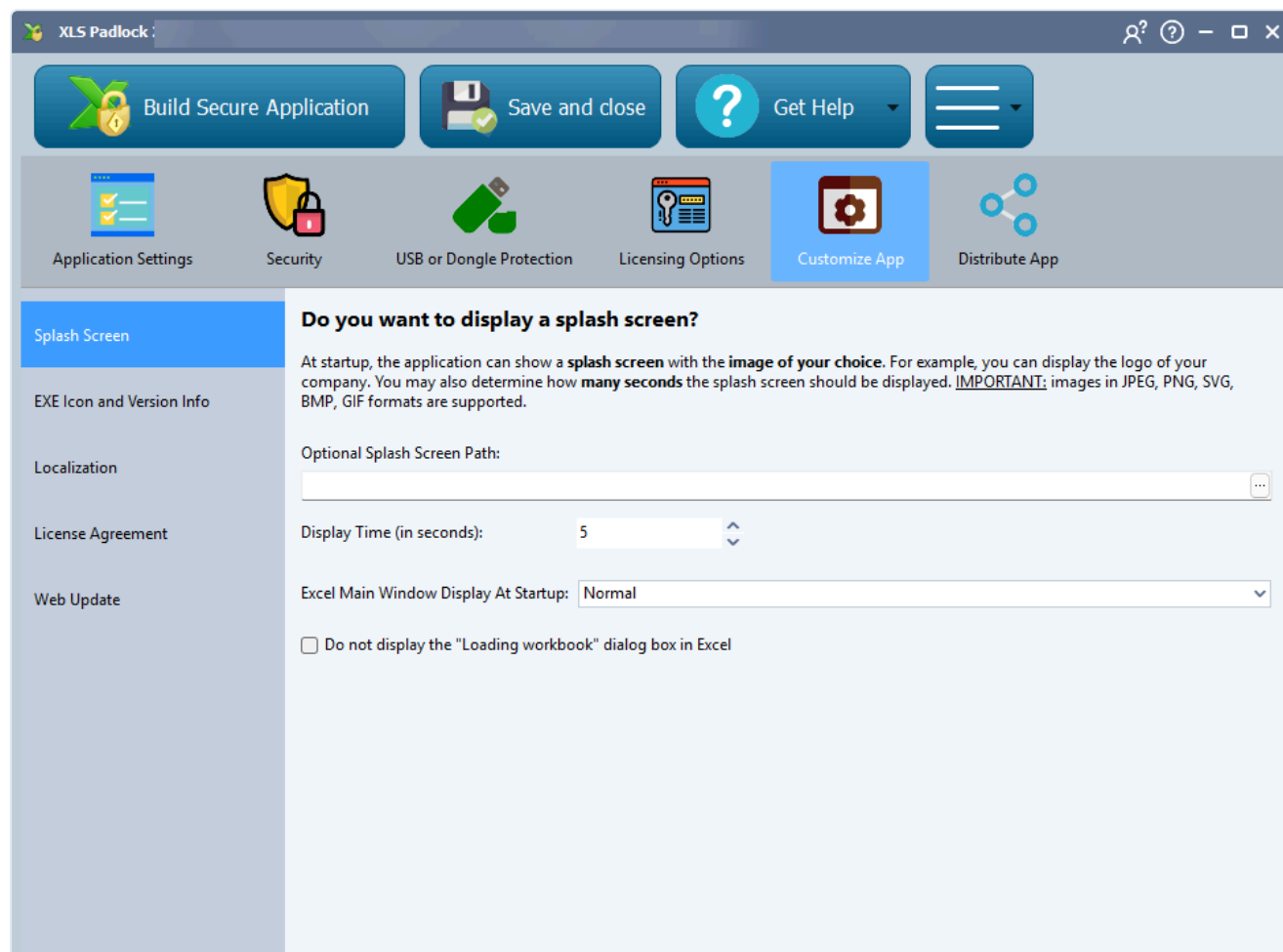
```
Sub Test_File()
    DoSomethingWith(PathToFile("data.xls"))
End Sub
```

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

Personalizar o aplicativo

O XLS Padlock oferece várias maneiras de **personalizar sua aplicação de Excel compilada**, permitindo que você adapte a aparência, as mensagens e o comportamento para os usuários finais.

Este tópico apresenta uma visão geral das opções disponíveis em "**Customize App**" no XLS Padlock:



Splash Screen

Exiba uma **splash screen** (tela de abertura) personalizada quando a sua aplicação é iniciada, por exemplo, o logotipo da sua empresa ou uma imagem de boas-vindas com a sua marca. Você pode escolher o formato da imagem (JPEG, PNG, BMP, SVG ou GIF), definir por quantos segundos ela permanece visível e controlar como a janela do Excel se comporta durante a inicialização.

[→ Saiba mais sobre as configurações de Splash Screen](#)

EXE Icon and Version Info

Substitua o ícone padrão do XLS Padlock pelo **ícone da sua própria aplicação**.

Você também pode especificar informações de versão, como **product name**, **company name**, **copyright** e **file description**, tudo visível na caixa de diálogo de propriedades de arquivo do Windows.

Isso ajuda você a dar ao seu EXE a imagem de uma aplicação independente e profissional.

→ [Saiba mais sobre EXE Icon e Version Info](#)

Localization

Personalize ou traduza todas as **caixas de diálogo, mensagens e avisos integrados** exibidos pela sua planilha protegida.

O XLS Padlock oferece um sistema de localização flexível para que você possa adaptar a sua aplicação a diferentes idiomas ou criar mensagens de usuário totalmente personalizadas.

→ [Saiba mais sobre Localization](#)

License Agreement

Adicione um **license agreement** (contrato de licença) que os usuários devem aceitar antes de executar pela primeira vez a sua aplicação de Excel compilada.

Você pode incluir texto sem formatação ou carregar um arquivo RTF/HTML externo.

Se os usuários recusarem o contrato, a aplicação não continuará a abrir.

→ [Saiba mais sobre License Agreement](#)

Web Update

Habilite as **atualizações via web** para a sua aplicação de Excel.

Este recurso permite que você notifique os usuários quando uma nova versão do seu EXE estiver disponível e que a baixe automaticamente do seu site ou servidor.

É uma forma eficiente de manter os seus usuários atualizados com o mínimo de esforço.

→ [Saiba mais sobre Web Update](#)

Empacotamento do aplicativo

A opção **Application Packaging** (empacotamento da aplicação) determina como sua planilha protegida é compilada, oferecendo dois formatos distintos.

Where do you want to create the secure application?

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

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

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

Output Path: Copy Browse... Open Folder

Application Title: i

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

Build EXE for Excel: v i

Recomendação

Recomendamos a opção **Standalone EXE** se você tiver um [certificado de assinatura digital](#). Embora não seja obrigatório, assinar seu EXE é a melhor forma de evitar os avisos de "Unknown App" do Windows SmartScreen e de reduzir os falsos positivos do software antivírus.

Standalone EXE

Esta opção compila sua planilha do Excel em um único arquivo executável (.exe) independente. Você pode distribuir esse arquivo único aos seus usuários, e eles não precisarão da planilha original do Excel.

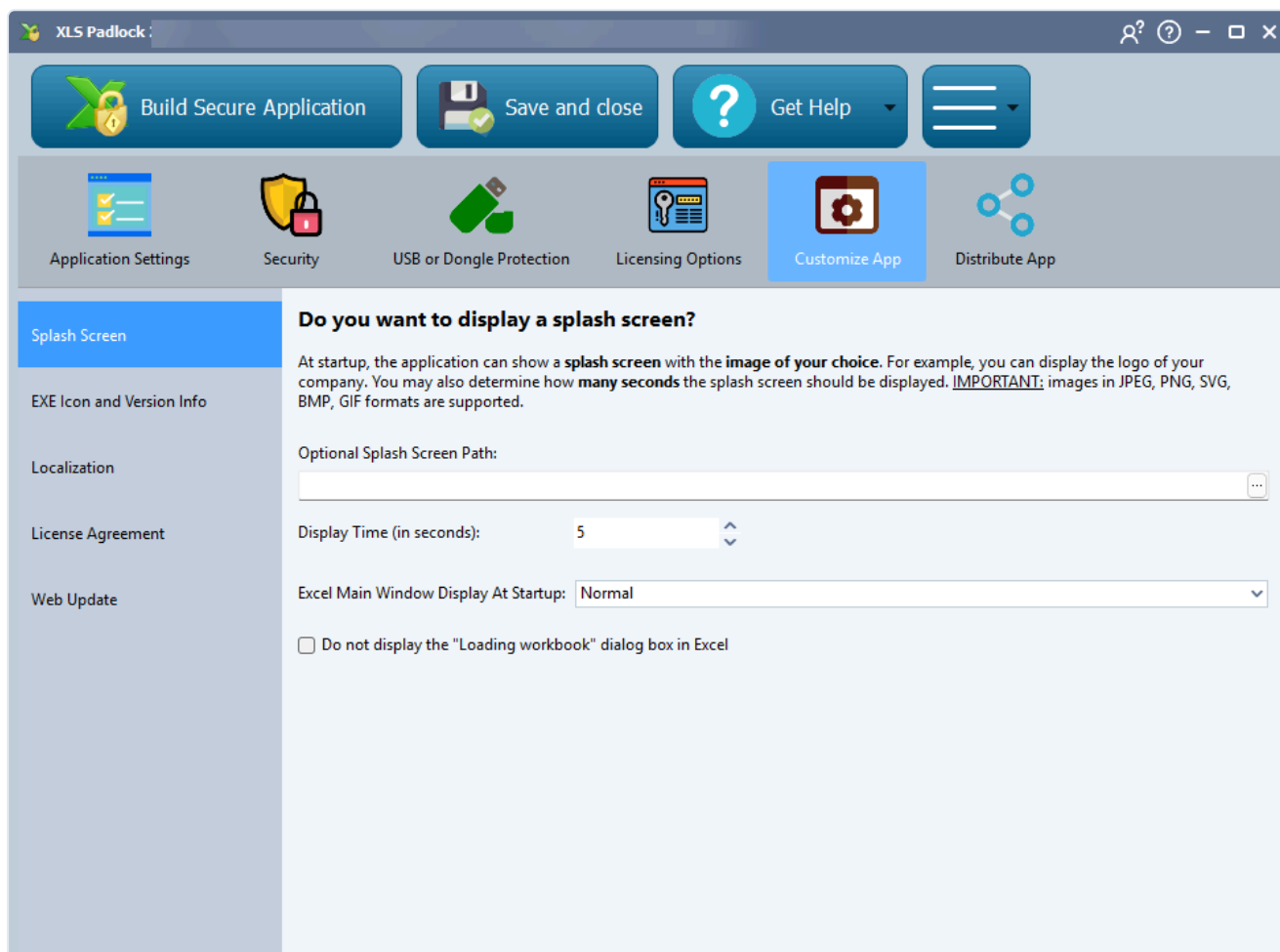
EXE + XPLAPP Application Bundle

Esta opção cria um arquivo EXE junto com um arquivo complementar `.bin64` e um arquivo de dados `.xplapp`. O arquivo EXE principal vem pré-assinado por nossa empresa, o que ajuda a torná-lo mais confiável para o Windows SmartScreen e para as soluções antivírus.

- Este método de distribuição requer que os arquivos EXE, `.bin64` e `.xplapp` sejam todos mantidos na mesma pasta.
- Para maior comodidade, o XLS Padlock pode agrupar esses arquivos em um [único arquivo Zip](#) ou em [um instalador](#).
- O uso deste método reduz consideravelmente as chances de encontrar alertas de "Unknown Application" e falsos positivos dos antivírus.
- O arquivo complementar `.bin64` garante a compatibilidade com as versões de 64 bits do Excel.

Tela de abertura

Você pode exibir uma **tela de abertura** (splash screen), uma imagem mostrada brevemente na inicialização, enquanto seu aplicativo está sendo inicializado. Essa é uma ótima maneira de exibir o logotipo da sua empresa ou de dar identidade de marca ao seu aplicativo.



Configurações da tela de abertura

Você pode usar qualquer imagem nos formatos JPEG, PNG, BMP, SVG ou GIF.

Você pode definir a duração pela qual a tela de abertura é exibida. Por padrão, os usuários podem fechá-la imediatamente clicando sobre ela. Esse comportamento pode ser desativado nas [Advanced Options](#).

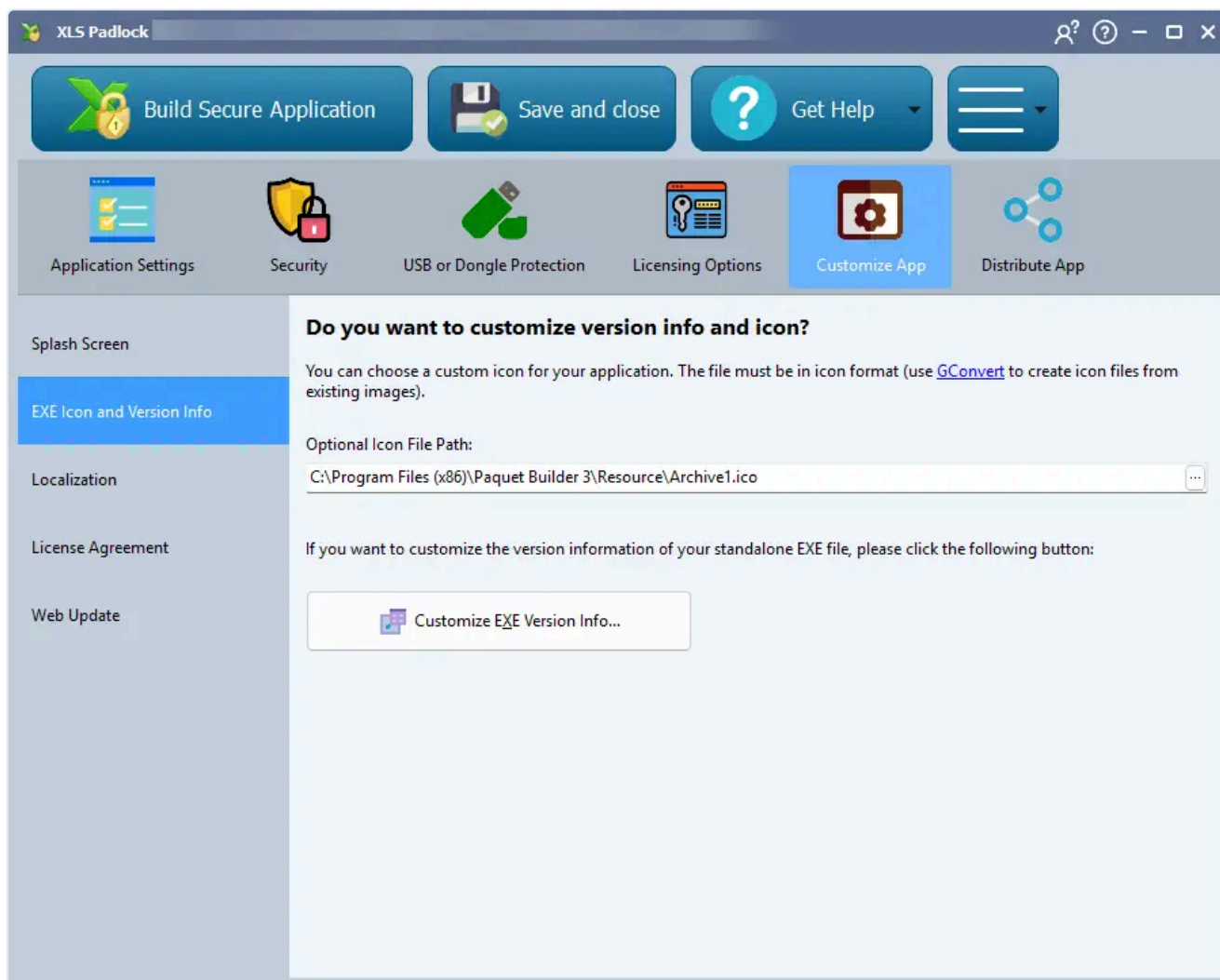
Use PNGs transparentes para um visual único

O XLS Padlock oferece suporte a telas de abertura não retangulares e semitransparentes. Use um arquivo PNG de 32 bits com canal alfa para dar ao seu aplicativo uma aparência moderna e personalizada.

👉 Consulte também: [Não exibir a caixa de diálogo "Loading workbook" no Excel](#chapter-do-not-display-the-loading-workbook-dialog-box).

Alterar o ícone do EXE

O XLS Padlock permite substituir o **ícone padrão da aplicação** por um próprio. Para personalizar a aparência da sua aplicação, especifique o **caminho para um arquivo de ícone personalizado**, que deve ter a extensão `.ico` :



O XLS Padlock oferece suporte a formatos de ícone padrão, incluindo diversos tamanhos (por exemplo, 32x32, 48x48) e profundidades de cor. Se você precisar de uma ferramenta para criar ou extrair ícones, recomendamos nosso utilitário complementar, [GConvert](#), que pode converter imagens em ícones com facilidade.

Aviso

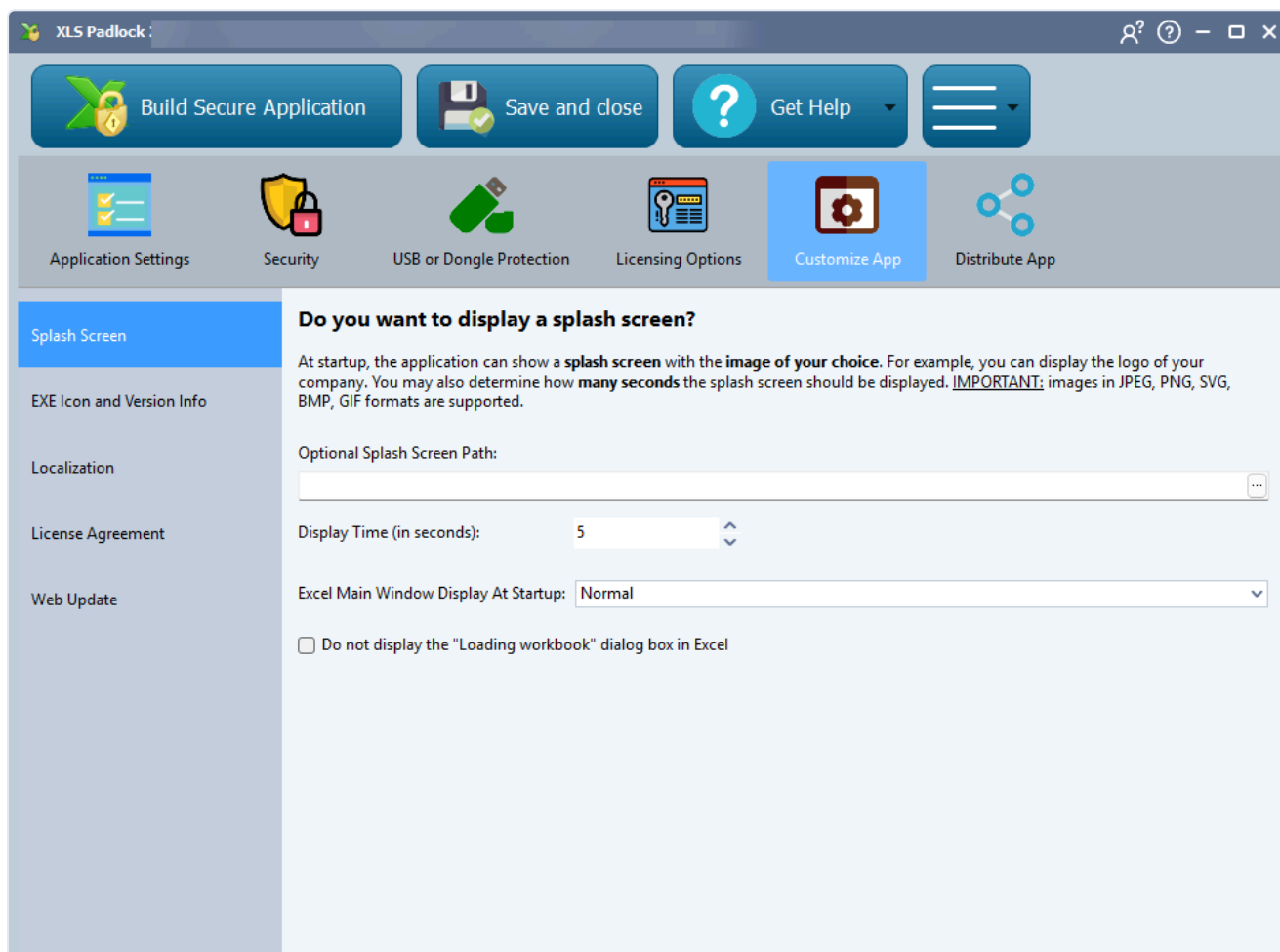
O arquivo de ícone deve estar disponível como um arquivo externo no seu computador durante o processo de compilação.

Informação

O XLS Padlock também aplicará este ícone ao arquivo EXE da aplicação independente, garantindo uma identidade de marca coerente.

Janela do Excel ao iniciar

A opção **Excel Main Window Display at Startup** (exibição da janela principal do Excel na inicialização) permite definir como a janela do Excel será dimensionada quando seu aplicativo for iniciado. Essa configuração fica na página [Splash Screen](#).



Você pode escolher entre três opções:

- **Normal:** a janela do Excel abre com seu tamanho e posição padrão. Esse é o comportamento padrão.
- **Minimized:** o aplicativo é iniciado minimizado na barra de tarefas do Windows. Isso é útil se seu aplicativo for executado em segundo plano ou se você quiser exibir um formulário personalizado antes de mostrar a janela principal do Excel.
- **Maximized:** a janela do Excel abre maximizada para preencher a tela inteira.

Precisa ocultar completamente a janela do Excel?

Se seu aplicativo for controlado apenas por código VBA e UserForms e você quiser que a janela do Excel, e seu ícone na barra de tarefas, nunca apareça, consulte [Executar como um aplicativo somente VBA \(janela principal do Excel totalmente oculta\)](#). Quando essa opção está ativada, a escolha acima é substituída e o Excel sempre é iniciado oculto.

Veja Também

- [Como configurar a Splash Screen](#)
- [Como ocultar a caixa de diálogo "Loading workbook"](#)
- [Executar como um aplicativo somente VBA \(janela principal do Excel totalmente oculta\)](#)

Executar como aplicativo somente VBA

Se a sua planilha protegida é controlada inteiramente por **código VBA e UserForms**, e a própria planilha nunca deve ser vista pelos seus usuários finais, você pode distribuí-la como um *aplicativo somente de VBA*. A janela principal do Excel permanece oculta do início ao fim, incluindo o seu ícone na barra de tarefas, e somente o seu UserForm aparece na tela. O seu aplicativo se parece e se comporta como um programa independente do Windows, sem nenhum rastro visível do Excel.

Esta opção está disponível a partir do **XLS Padlock 2026.0**.

Quando utilizá-la

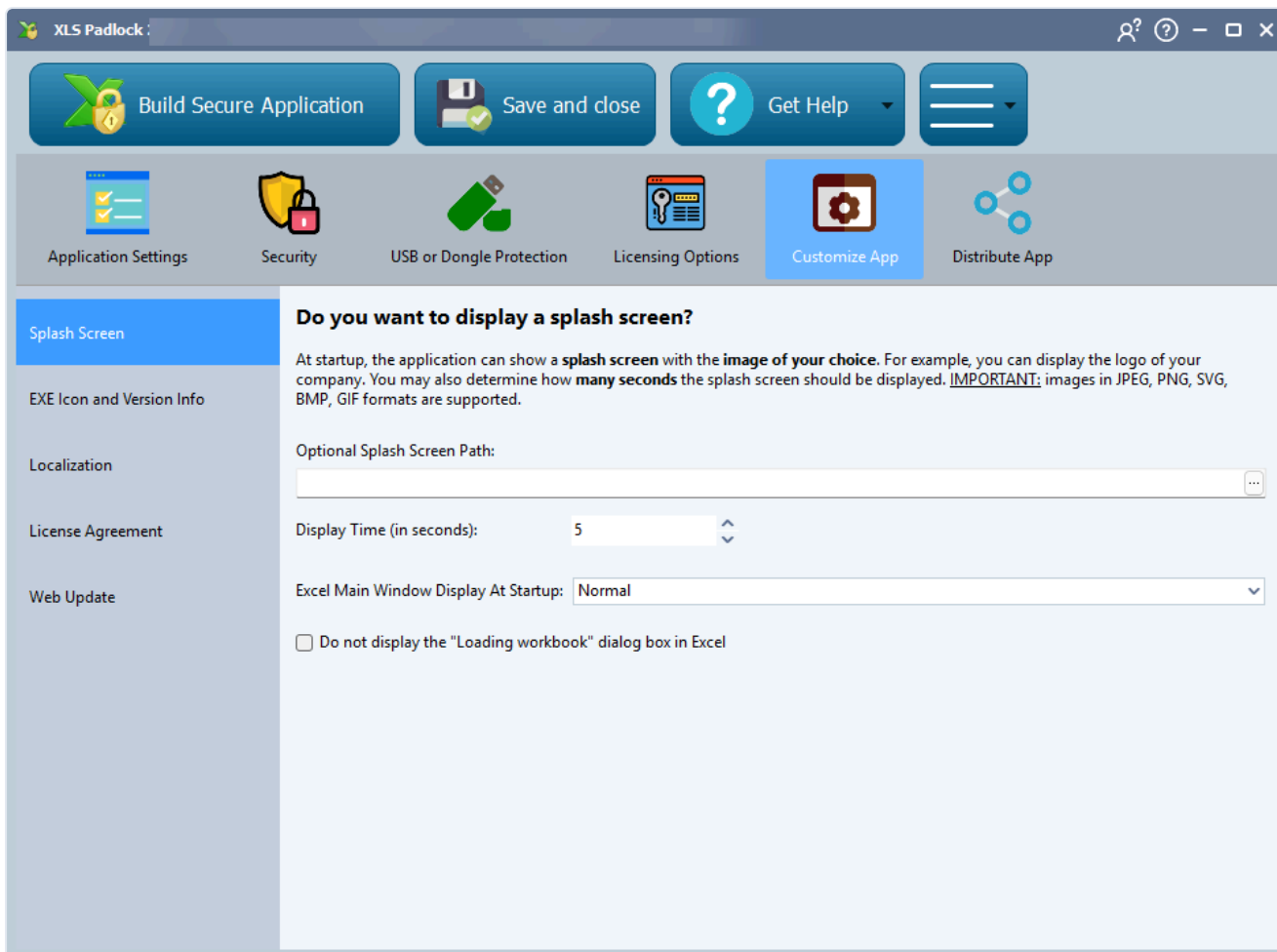
Utilize o modo somente de VBA quando **todas** as condições a seguir forem verdadeiras:

- A interface de usuário do seu aplicativo é construída com **UserForms**, não com planilhas.
- Você não quer que os usuários finais vejam, editem ou interajam com a planilha subjacente.
- Você deseja uma aparência limpa e com a sua marca na inicialização, sem tela de apresentação do Excel, sem entrada do Excel na barra de tarefas.

Se o seu aplicativo depende de planilhas visíveis (grades de entrada de dados, painéis renderizados como células, etc.), **não** ative esta opção, pois as planilhas não ficarão visíveis para os seus usuários.

Como ativá-la

1. Abra o seu projeto no XLS Padlock.
2. Vá até a página **Splash Screen** (em Application Customization).
3. Marque a opção **"Run as a VBA-only app (Excel main window fully hidden)"**.



E isso por parte do XLS Padlock. Agora você precisa adicionar um pequeno trecho de código VBA à sua planilha para que o seu UserForm seja exibido quando o aplicativo iniciar.

Adicionar o código VBA de inicialização

No editor de VBA do Excel, abra o módulo `ThisWorkbook` e cole:

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

Substitua `UserForm1` pelo nome do formulário que você deseja exibir primeiro.

Por que ambos, a caixa de seleção *e* o código VBA?

A caixa de seleção do XLS Padlock garante que a janela principal do Excel nunca se torne visível **antes** que o seu VBA seja executado (sem piscar, sem ícone na barra de tarefas). A linha `Application.Visible = False` em `Workbook_Open` mantém o Excel oculto **depois** que o Excel termina de inicializar, já que o Excel restabelece a sua visibilidade internamente durante a inicialização, portanto o VBA precisa impor a regra assim que a planilha estiver totalmente carregada. Usar ambas as camadas proporciona uma experiência de usuário limpa e sem cintilação.

O que o usuário final vê

Com a caixa de seleção ativada e o trecho de VBA acima:

- **Sem tela de apresentação do Excel** na inicialização.
- **Sem janela principal do Excel** em nenhum momento durante a vida do aplicativo.
- **Sem entrada de "Excel" na barra de tarefas do Windows**, apenas o seu UserForm.
- O seu aplicativo se comporta como um programa independente do Windows construído em torno do seu formulário.

Se você também configurou uma [splash screen](#) e ocultou a [caixa de diálogo Loading workbook](#), toda a sequência de inicialização permanece dentro da sua interface com a sua marca desde o primeiro quadro.

Fechar o aplicativo

Como o Excel está oculto, o botão padrão de fechar do Excel não fica disponível para os seus usuários. Normalmente, você fecha o aplicativo a partir do seu UserForm usando:

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

Ou, se você quiser deixar o Excel em execução, mas descarregar a sua interface:

```
Unload Me
```

Interação com outras configurações

Configuração	Comportamento quando "VBA-only app" está ativado
Excel Main Window Display at Startup (Normal / Maximized / Minimized)	Substituído, o Excel sempre inicia oculto.

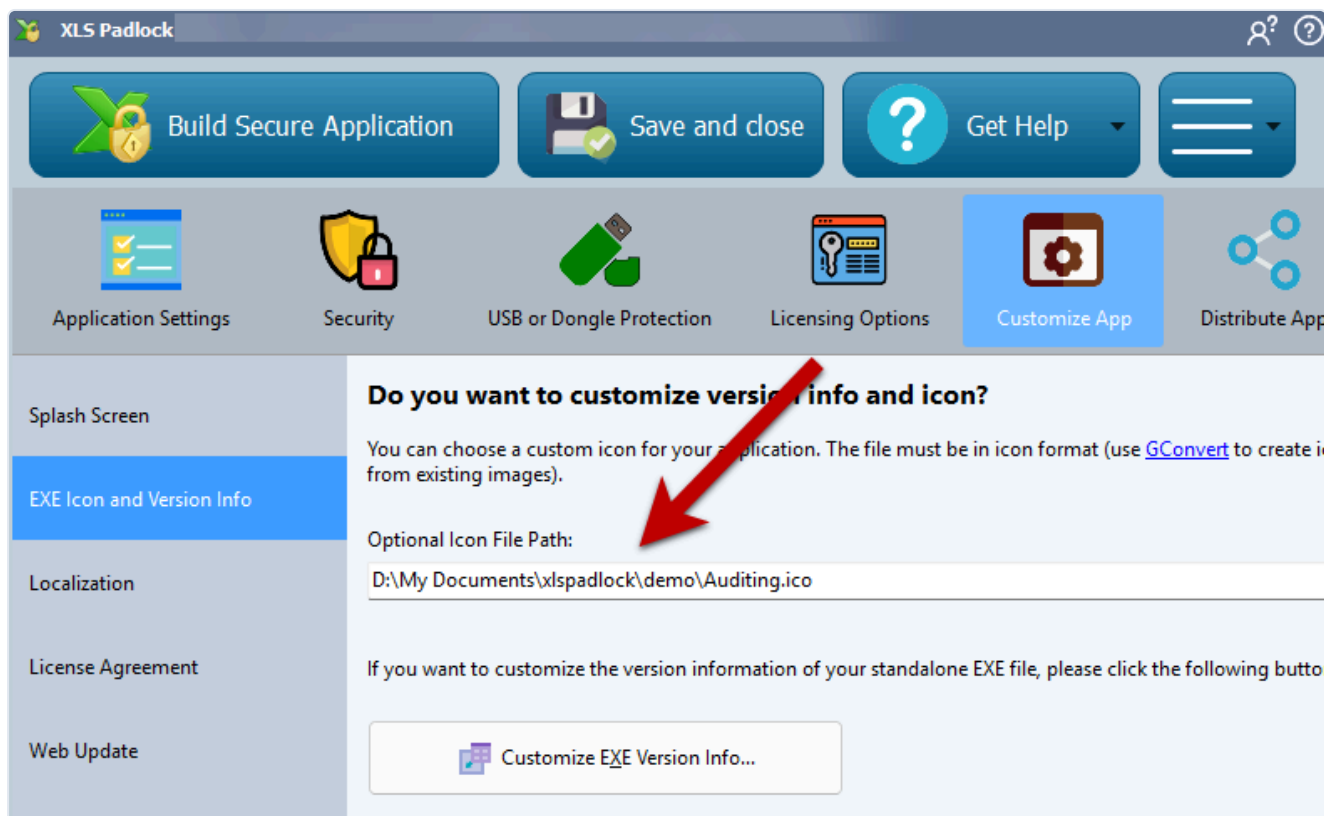
Configuração	Comportamento quando "VBA-only app" está ativado
Do not display the "Loading workbook" dialog box	Recomenda-se ativar também, para uma inicialização totalmente silenciosa.
Splash Screen	Funciona conforme configurado. Exibida antes mesmo de o Excel carregar.

Veja também

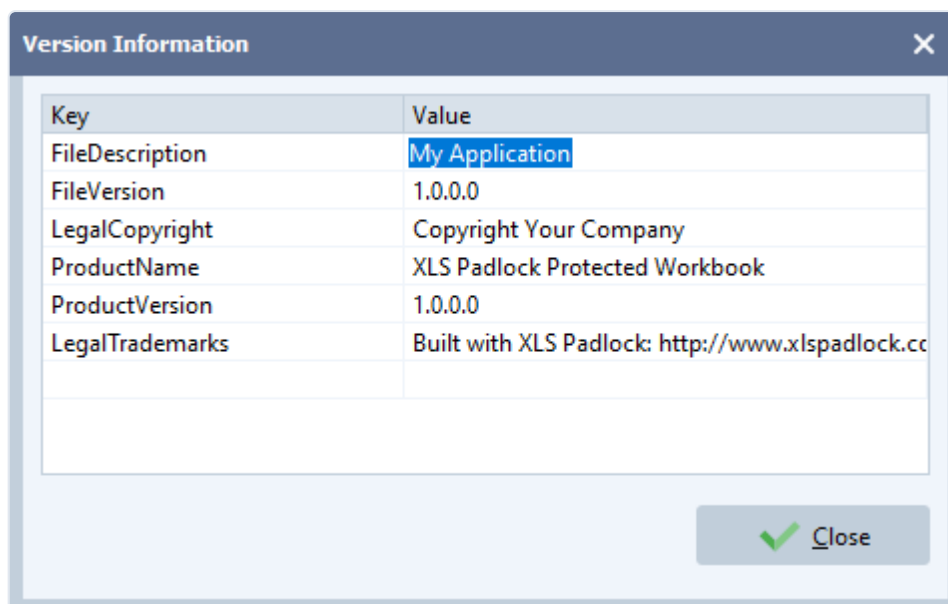
- [Como configurar a Splash Screen](#)
- [Como definir a exibição da janela principal do Excel na inicialização](#)
- [Como ocultar a caixa de diálogo "Loading workbook"](#)
- [Livro de receitas da API de VBA](#)

Informações de versão do EXE

As **informações de versão** de um arquivo executável são uma seção de recursos especial que contém detalhes como o número de versão do arquivo, o sistema operacional pretendido, o nome de arquivo original e as informações de copyright. Esses dados são incorporados ao EXE compilado. Quando incluídos, os usuários podem visualizar essas informações clicando com o botão direito no ícone do programa, selecionando "Properties" (Propriedades) e navegando até a aba "Details" (Detalhes), ou pressionando **Alt+Enter** sobre o arquivo no Explorador.



O XLS Padlock permite que você incorpore as suas próprias informações de versão no arquivo EXE independente. Clique em "**Customize EXE Version Info**" (Personalizar informações de versão do EXE) para abrir uma janela com os seguintes campos:



- **File Description:** uma breve descrição do conteúdo da sua aplicação.
- **Company Name:** o nome da sua empresa.
- **File Version:** o número de versão do seu arquivo `.exe`, no formato `X.X.X.X` (por exemplo, `1.20.34.45`). Esse valor também é usado pela [função Web Update](#) para verificar se há novas versões da aplicação.
- **Legal Copyright:** o seu aviso de copyright, como "Copyright © [annee] Your Company. All rights reserved."
- **Product Name:** o nome do seu produto ou aplicação, que geralmente é o mesmo que o título da sua aplicação.
- **Product Version:** o número de versão do seu produto. Muitas vezes coincide com a File Version.
- **Legal Trademarks:** quaisquer marcas comerciais legais que você deseje incluir.

Informação

A File Version é usada pela [função Web Update](#).

Opções de linha de comando

O arquivo EXE do seu aplicativo compilado oferece suporte a vários **modificadores de linha de comando** (também conhecidos como argumentos) que automatizam ações específicas na inicialização. Você pode usá-los em atalhos ou scripts para controlar como o aplicativo se comporta ao ser iniciado.

DICA

👉 Esses argumentos de linha de comando também podem ser recuperados em tempo de execução por meio da [API do VBA para ler parâmetros de linha de comando](#).

Modificadores disponíveis

-deact

Inicia o processo de desativação, permitindo que um usuário transfira sua licença para outro computador.

```
MYAPP.EXE -deact
```

-deL

Apaga todos os arquivos de salvamento seguro do aplicativo e carrega a planilha original. **Use com cautela, pois esta ação não pode ser desfeita.**

```
MYAPP.EXE -deL
```

-enterkey

Abre a caixa de diálogo de ativação, permitindo que o usuário insira uma nova chave de ativação. Isso é útil para substituir uma chave antiga ou expirada.

```
MYAPP.EXE -enterkey
```

-load

Abre uma caixa de diálogo de arquivos que solicita ao usuário que selecione um arquivo de salvamento para carregar.

```
MYAPP.EXE -load
```

-reset

Descarta quaisquer alterações não salvas da última sessão e carrega a planilha original, sem modificações. Isso não exclui nenhum arquivo de salvamento `.xlsc`.

```
MYAPP.EXE -reset
```

-webupdate

Força o aplicativo a verificar imediatamente se há uma nova versão on-line, com base nas suas configurações de "Web Update".

```
MYAPP.EXE -webupdate
```

Carregar um arquivo de salvamento específico

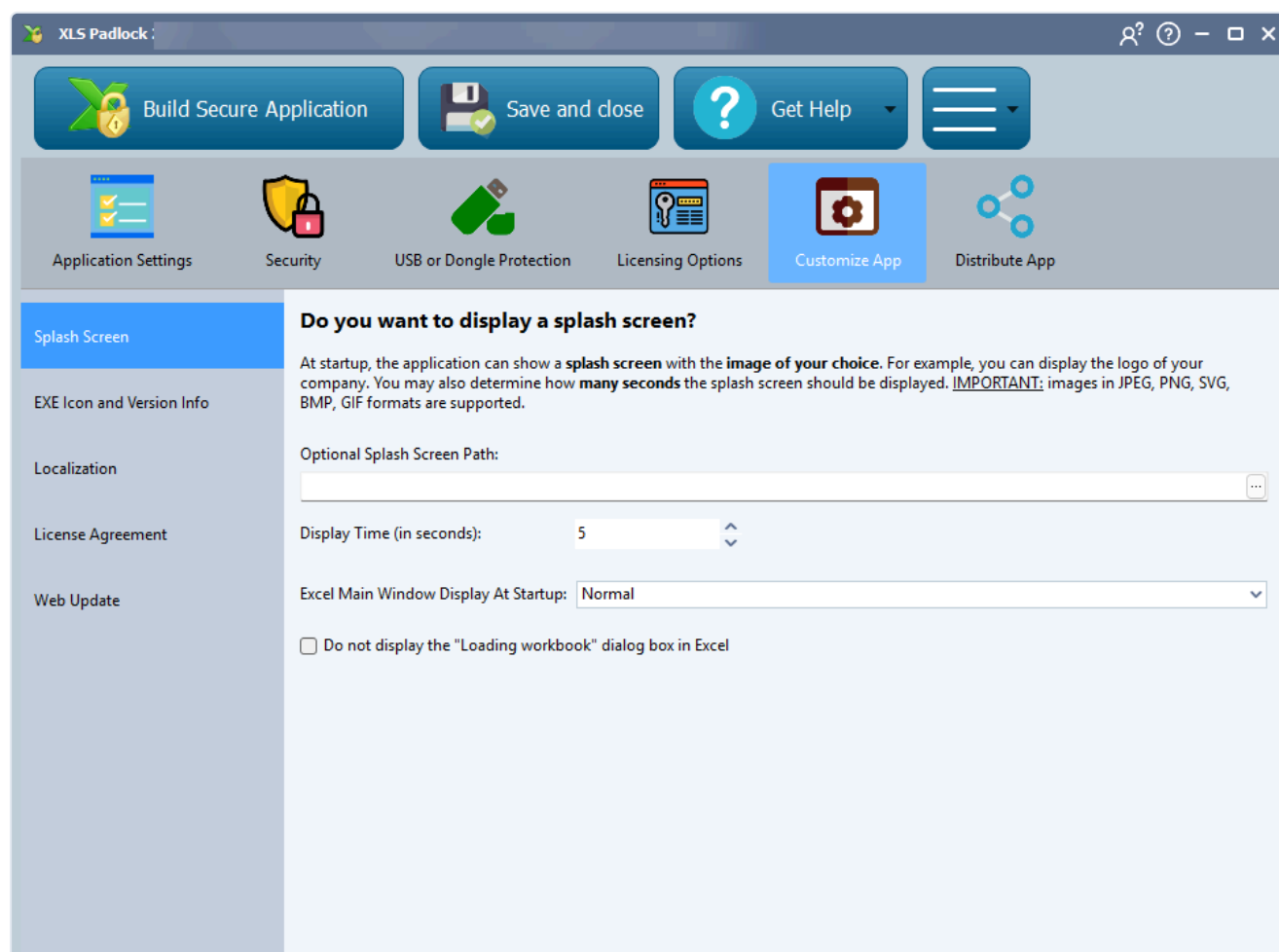
Você também pode iniciar o aplicativo com o caminho completo de um arquivo de salvamento seguro (`.xlsc` ou `.xlsce`) como argumento. Isso carregará automaticamente o arquivo especificado na inicialização, sem nenhuma solicitação.

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

Ocultar a caixa de diálogo de carregamento

Por padrão, quando sua aplicação protegida é iniciada, o Excel exibe uma pequena caixa de diálogo com a mensagem **"Loading workbook, please wait..."**.

Você pode ocultar essa caixa de diálogo habilitando a opção **"Do not display the "Loading workbook" dialog box in Excel"**, localizada na página [Splash Screen](#). Isso é útil para criar uma experiência de inicialização mais limpa, especialmente quando você já está usando uma tela de abertura personalizada.



Também é possível [ocultar essa mensagem de forma antecipada usando código VBA](#), o que lhe dá mais controle sobre quando a caixa de diálogo desaparece.

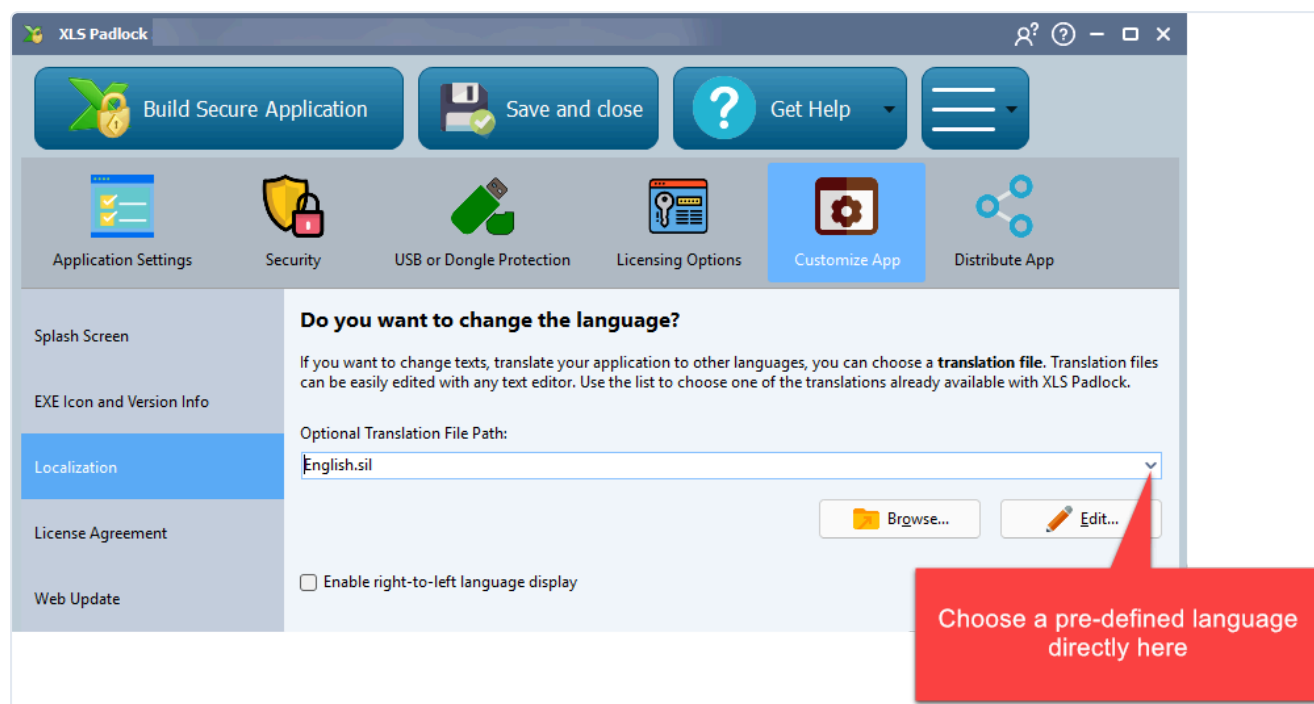
👉 Consulte também

- [Como configurar a Splash Screen](#)
- [Como definir a exibição da janela principal do Excel na inicialização](#)

Localização e tradução

O XLS Padlock oferece **suporte a localização** para suas planilhas protegidas, permitindo exibir todo o texto da interface (títulos de diálogo, mensagens, solicitações e avisos em tempo de execução) no idioma de sua escolha.

Você pode configurar a localização da sua planilha protegida aqui:



Formato do arquivo de tradução

Desde o XLS Padlock 2026, os arquivos de tradução usam o formato **GNU gettext padrão** com a extensão `.po` (e sua contraparte binária compilada `.mo`). É o mesmo formato usado pelas distribuições Linux, Python, PHP, WordPress e muitos outros aplicativos, o que significa que você pode editar as traduções com qualquer uma das dezenas de ferramentas gettext consolidadas disponíveis, incluindo o editor gratuito e multiplataforma [Poedit](#).

O aplicativo compilado incorpora um único recurso `.mo` no momento da compilação, carregado inteiramente da memória em tempo de execução. Nenhum arquivo temporário é extraído para o disco.

O XLS Padlock inclui arquivos `.po` já traduzidos para **10 idiomas**, disponíveis na subpasta `locale\` do diretório de instalação do XLS Padlock:

Arquivo	Idioma
<code>en.po</code>	Inglês
<code>fr.po</code>	Francês
<code>es.po</code>	Espanhol
<code>pt.po</code>	Português (Brasil)

Arquivo	Idioma
nL.po	Holandês
de.po	Alemão
ar.po	Árabe (com suporte a layout da direita para a esquerda)
it.po	Italiano
zh_CN.po	Chinês simplificado
zh_TW.po	Chinês tradicional

Solicitar um novo idioma

Se precisar de um idioma que não esteja na lista, [entre em contato conosco](#). Você também pode criar o seu próprio arquivo `.po` a partir de qualquer um dos anteriores como ponto de partida. Consulte *Edição de arquivos de tradução* mais abaixo.

Como definir um idioma para sua planilha protegida

- Na página **Localization**, clique em **Browse** (Procurar) ao lado do campo *Translation file* (arquivo de tradução). A caixa de diálogo abre por padrão na pasta `locale\` da sua instalação do XLS Padlock, onde ficam os 10 arquivos `.po` prontos. Escolha o idioma desejado ou navegue até um arquivo `.po` personalizado seu.
- Você também pode colar o caminho completo manualmente no campo *Translation file*, ou deixá-lo vazio para usar a tradução inglesa padrão incorporada ao XLS Padlock.

IMPORTANTE

O arquivo de tradução selecionado deve estar acessível no caminho especificado **quando você compilar seu aplicativo**. O XLS Padlock lê e compila esse arquivo durante o processo de compilação e incorpora o recurso `.mo` resultante diretamente no EXE protegido. Não há dependência em tempo de execução do arquivo `.po` original.

Edição de arquivos de tradução

Os arquivos `.po` são arquivos de texto UTF-8 simples. Você pode editá-los de duas maneiras:

- **Com o Poedit** (poedit.net, gratuito, multiplataforma), a opção recomendada. Ele mostra as cadeias de origem e traduzidas lado a lado, sinaliza as entradas duvidosas (fuzzy) e não traduzidas, valida a sintaxe ao salvar e produz uma saída UTF-8 limpa.
- **Com qualquer editor de texto** (Notepad++, VS Code, ...). Salve como **UTF-8 sem BOM** e mantenha um par `msgid / msgstr` por bloco. As linhas que começam com `#` são comentários e não são exibidas ao usuário final.

Recomendamos que você **copie um dos arquivos .po prontos** para um novo local (ao lado da sua planilha, por exemplo), modifique a cópia e, em seguida, aponte o XLS Padlock para o seu novo arquivo.

Modificar arquivos dentro da pasta de instalação do XLS Padlock geralmente requer direitos administrativos e pode ser sobrescrito por uma atualização futura.

Migração dos arquivos de tradução `.sil` legados

Se você personalizou a interface de usuário da sua planilha protegida no XLS Padlock 2025.3 ou anterior, a sua tradução foi armazenada como um arquivo `.sil` (o formato de tradução legado baseado em texto usado pelas versões anteriores). O XLS Padlock 2026 não lê mais arquivos `.sil` no momento da compilação. O formato foi substituído pelo `.po`.

Para evitar redigitar qualquer coisa, a distribuição do XLS Padlock inclui um pequeno **utilitário de migração SIL → PO** em `tools\sil_to_po\` na árvore de fontes:

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

Isso produz `customer_fr.po` na pasta atual, automaticamente mesclado com o `.po` canônico atual, de modo que:

- a coluna correspondente ao seu idioma de destino é copiada,
- as cadeias adicionadas desde a criação do seu `.sil` aparecem com um `msgstr` vazio (prontas para preencher no Poedit),
- as cadeias cuja origem inglesa mudou ligeiramente entre versões são sinalizadas como `#, fuzzy` para a sua revisão,
- as cadeias que não existem mais no XLS Padlock são acrescentadas ao final como entradas obsoletas `#~` (ou descartadas por completo com `--drop-orphans`).

Para migrar vários idiomas em uma única execução, separe-os por vírgula:

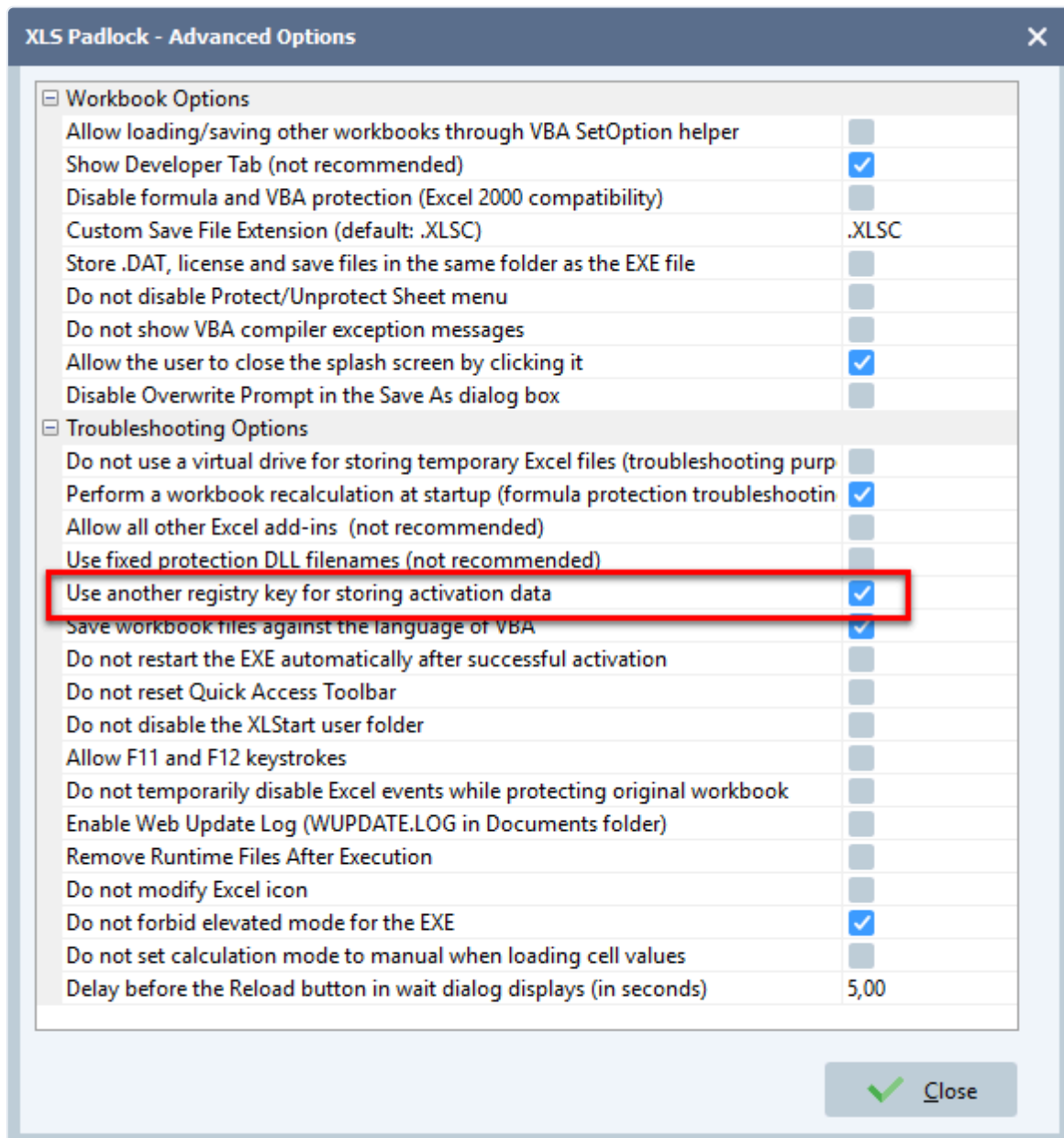
```
python tools/sil_to_po/sil2po_customer.py your_translation.sil --lang fr,es,de --stats
```

Os arquivos `.po` resultantes são imediatamente utilizáveis a partir do seletor *Translation file* no XLS Padlock Designer. O script utilitário e seu `README.md` complementar (referência completa de opções, algoritmo de correspondência, exemplos resolvidos) são fornecidos na subpasta `tools\sil_to_po\` do diretório de instalação do XLS Padlock. Abra o `README.md` em qualquer editor de texto para ler a documentação completa localmente.

É necessário o Python 3.8 ou posterior para executar o utilitário. Sem dependências externas de Python: o script é autônomo.

Opções avançadas

Em “Application Settings”, clique no botão “Configure Advanced Options” para exibir a seguinte janela:



Apenas para usuários avançados

Estas opções destinam-se a usuários avançados. Altere-as somente se você compreender a finalidade delas ou se for orientado pelo nosso suporte técnico.

Opções avançadas disponíveis

- **Allow loading/saving other workbooks through VBA SetOption helper:** uma omissão de segurança para operações específicas de VBA. Consulte [Carregar/salvar planilhas por meio do auxiliar VBA SetOption](#).

- **Show Developer Tab:** torna visível a guia Developer (Desenvolvedor) do Excel (não recomendado por motivos de segurança).
- **Disable formula and VBA protection:** para compatibilidade com o Excel 2000.
- **Custom Save File Extension:** altera a extensão padrão `.XLSC` dos arquivos salvos.
- **Store .DAT, license and save files in the same folder as the EXE file:** habilita o modo portátil.
- **Do not disable Protect/Unprotect Sheet menu:** mantém habilitados os itens de menu de proteção de planilhas.
- **Do not show VBA compiler exception messages:** oculta as mensagens de erro detalhadas do VBA Compiler. Consulte [Desativar as informações de depuração](#).
- **Allow the user to close the splash screen by clicking it:** permite que os usuários pulem a tela de abertura (splash screen).
- **Disable Overwrite Prompt in the Save As dialog box:** evita a mensagem "do you want to overwrite?".
- **Do not use a virtual drive for storing temporary Excel files:** para fins de resolução de problemas.
- **Perform a workbook recalculation at startup:** para resolver problemas de proteção de fórmulas.
- **Allow all other Excel add-ins:** reduz a segurança ao habilitar todos os suplementos (não recomendado).
- **Use fixed protection DLL filenames:** não recomendado por motivos de segurança.
- **Use another registry key for storing activation data:** uma solução alternativa para conflitos no registro.
- **Save workbook files against the language of VBA:** resolve alguns problemas de localização.
- **Do not restart the EXE automatically after successful activation:** evita a reinicialização automática.
- **Do not reset Quick Access Toolbar:** preserva as personalizações do usuário na QAT (Barra de Ferramentas de Acesso Rápido).
- **Do not disable the XLStart user folder:** permite a execução dos suplementos da pasta XLStart do usuário.
- **Allow F11 and F12 keystrokes:** habilita essas teclas de função.
- **Do not temporarily disable Excel events while protecting original workbook:** para fins de resolução de problemas.
- **Enable Web Update Log:** cria um arquivo `WUPDATE.LOG` na pasta Documentos do usuário para depurar o recurso de atualização via web.
- **Remove Runtime Files After Execution:** limpa os arquivos temporários.
- **Do not modify Excel icon:** evita que o XLS Padlock altere o ícone da janela do Excel.
- **Do not forbid elevated mode for the EXE:** permite que a aplicação seja executada como administrador.
- **Do not set calculation mode to manual when loading cell values:** para fins de resolução de problemas.
- **Delay before the Reload button in wait dialog displays (in seconds):** personaliza a caixa de diálogo de espera.

Ignorar erros de processamento

A opção "Ignore errors when processing the workbook (internal protection)" (ignorar erros ao processar a planilha, proteção interna) permite que o processo de compilação continue mesmo que sejam encontrados erros ao abrir a sua planilha do Excel. Esse recurso foi projetado para agilizar o fluxo de trabalho de proteção, suprimindo as mensagens de erro que, de outra forma, poderiam interromper o processo.

Quando essa opção está habilitada, o XLS Padlock não interrompe nem exibe mensagens de erro caso encontre problemas durante a fase de abertura da planilha.

Esta opção não corrige erros

Habilitar esta opção não resolve nenhum problema subjacente da planilha; ela apenas os ignora durante a compilação. Após proteger a sua planilha com esta opção habilitada, você deve testá-la minuciosamente para garantir que todos os recursos funcionem como esperado.

Se você continuar a encontrar erros na planilha compilada, considere habilitar a opção alternativa, "[Use Excel automation for formula protection](#chapter-formula-protection-method)", que pode atenuar problemas que o método padrão de proteção interna não resolve.

Desativar informações de depuração

Por padrão, se o VBA Compiler detectar um erro no código compilado em tempo de execução, ele exibe uma mensagem de erro detalhada para ajudar você a depurar o problema.

Se você não quiser exibir essas informações aos seus usuários finais, pode desativá-las. Vá até [Configurar opções avançadas](#) e desative a opção: **“Do not show VBA compiler exception messages”**.

Interface personalizada do Excel

Conforme explicado na documentação da Microsoft, você pode personalizar a interface de usuário do Excel colocando um arquivo `.officeUI` criado especialmente em uma pasta de usuário específica.

O XLS Padlock permite que você especifique seu próprio Excel.officeUI para a aplicação segura. Ele substituirá automaticamente a interface padrão do usuário (se disponível).

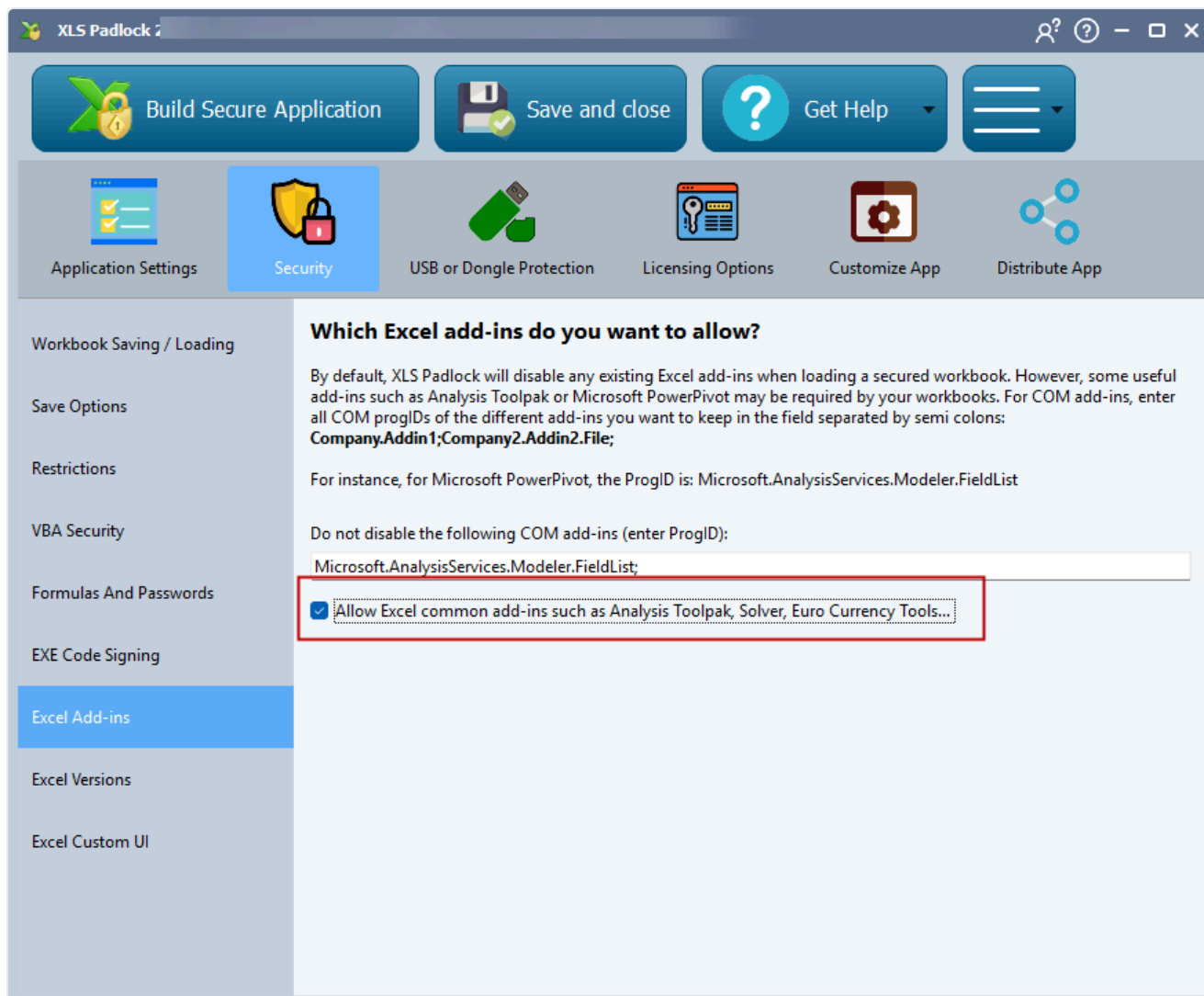
Basta fornecer o caminho completo para o seu arquivo `.officeUI`. O XLS Padlock o compilará diretamente no EXE final, de modo que você não precisa distribuir o arquivo `.officeUI` separadamente.

Para obter mais informações sobre a criação de arquivos XML de interface de usuário personalizada, consulte a [documentação oficial do Microsoft Office](#).

Suplementos do Excel

Por padrão, o XLS Padlock desativa a maioria dos suplementos do Excel quando uma planilha protegida é carregada. Esta é uma medida de segurança para criar um ambiente controlado para a sua aplicação.

No entanto, se a sua planilha depende de suplementos específicos para funcionar corretamente, você pode reativá-los de forma seletiva. Esta página explica como permitir tanto os suplementos COM quanto os suplementos integrados comuns do Excel.



Manter ativados suplementos COM específicos

Se a sua planilha exigir um suplemento COM (como o Microsoft PowerPivot), você deve especificar seu ProgID para impedir que o XLS Padlock o desative.

No campo "Do not disable the following COM add-ins" (Não desativar os seguintes suplementos COM), insira o ProgID de cada suplemento que deseja manter ativado. Se houver mais de um, separe-os com ponto e vírgula (;).

Exemplo: Para manter o Microsoft PowerPivot ativado, você inseriria seus ProgIDs:

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

Como encontrar um ProgID

O ProgID (identificador programático) é um nome único armazenado no registro do Windows para um componente COM. Muitas vezes você pode encontrá-lo na documentação do suplemento ou inspecionando o registro. A Microsoft fornece mais informações sobre este assunto em seu site de suporte.

Permitir suplementos comuns do Excel

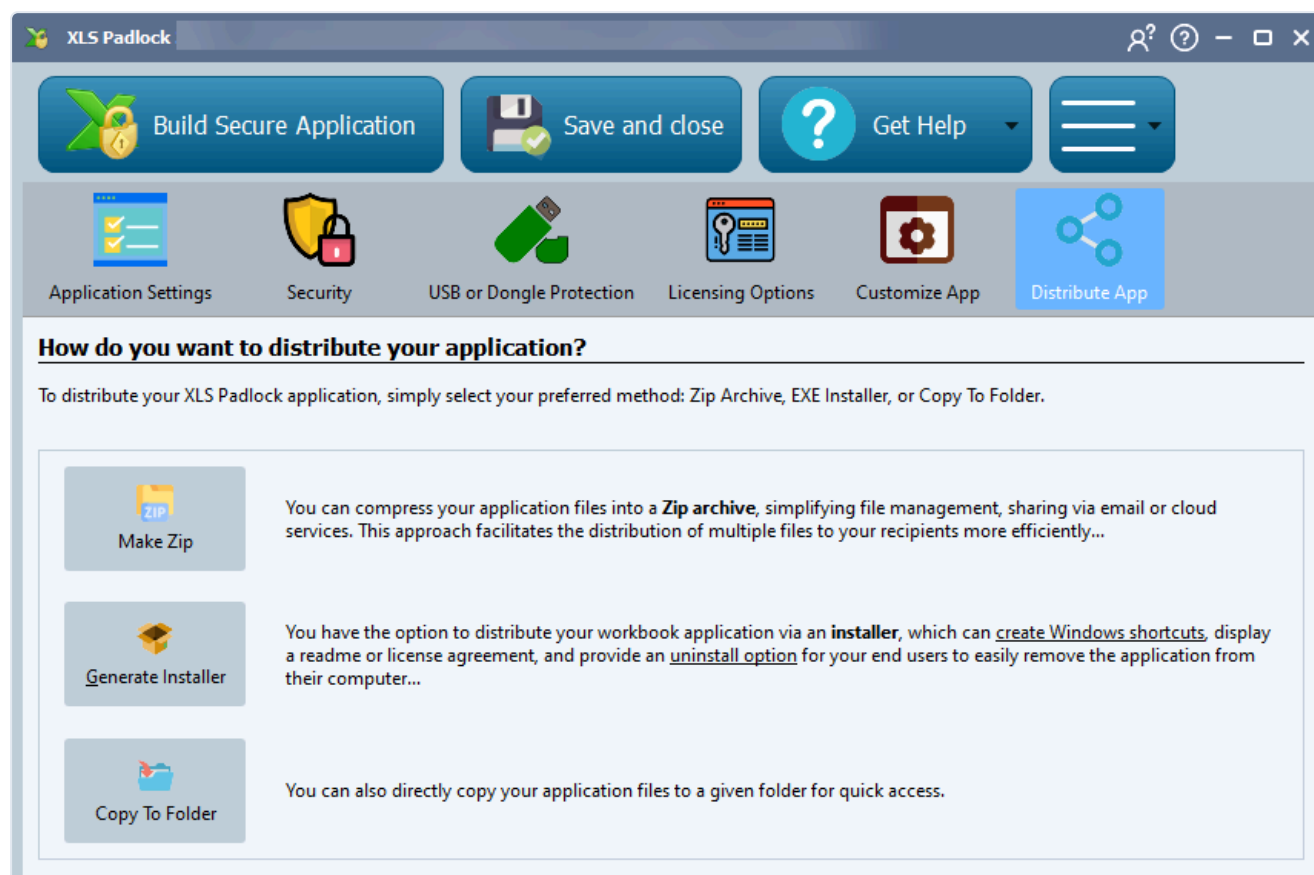
O Excel também inclui vários suplementos integrados populares, como o **Analysis ToolPak**, o **Solver** e as **Euro Currency Tools**.

Se a sua planilha utiliza recursos de qualquer um deles, ative a opção "**Allow Excel common add-ins...**" (Permitir suplementos comuns do Excel...). Isso garantirá que sejam carregados e estejam disponíveis quando a sua aplicação protegida for iniciada.

Distribuir o aplicativo

Depois de [converter sua planilha em um executável](#) (.EXE), o próximo passo é distribuí-la aos seus usuários.

O XLS Padlock oferece diversos métodos para agilizar esse processo: criar um arquivo Zip, gerar um instalador EXE ou copiar diretamente os arquivos para uma pasta.



Arquivo Zip

Um arquivo Zip é um método popular e simples para distribuir software. Comprimir sua aplicação em um único arquivo `.zip` simplifica o gerenciamento de arquivos para seus usuários. Esse método é especialmente útil para compartilhar sua aplicação por e-mail ou serviços em nuvem, pois reduz o tamanho do arquivo e reúne todos os componentes necessários em um único pacote.

Para criar um arquivo Zip da sua aplicação, clique em **Make Zip** (Criar Zip) e escolha onde deseja salvar o arquivo `.zip`.

Instalador EXE

Para um toque mais profissional, considere distribuir sua aplicação com um instalador EXE. Esse método aprimora a experiência do usuário ao oferecer um processo de instalação familiar, comum nos ambientes

Windows. Um instalador pode criar atalhos na área de trabalho e no menu Iniciar, exibir um arquivo leia-me ou um contrato de licença e fornecer um desinstalador, permitindo que os usuários removam facilmente a aplicação do computador.

➡ Para obter instruções detalhadas, consulte nosso guia sobre [como criar um instalador para sua aplicação](#).

Copiar para uma pasta

Por fim, copiar os arquivos da sua aplicação diretamente para uma pasta é o método de distribuição mais direto. Ele envolve dois passos simples:

- Copie todos os arquivos necessários da aplicação para uma pasta.
- Compartilhe essa pasta com seus usuários por meio de mídia física (como uma unidade USB) ou por transferência via rede.

Embora não seja tão refinado quanto os outros métodos, essa abordagem é a forma mais rápida de colocar sua aplicação em funcionamento em outro sistema, pois não requer utilitários zip especiais nem software de instalação.

👉 Consulte também:

- [Como assinar digitalmente sua aplicação](#)

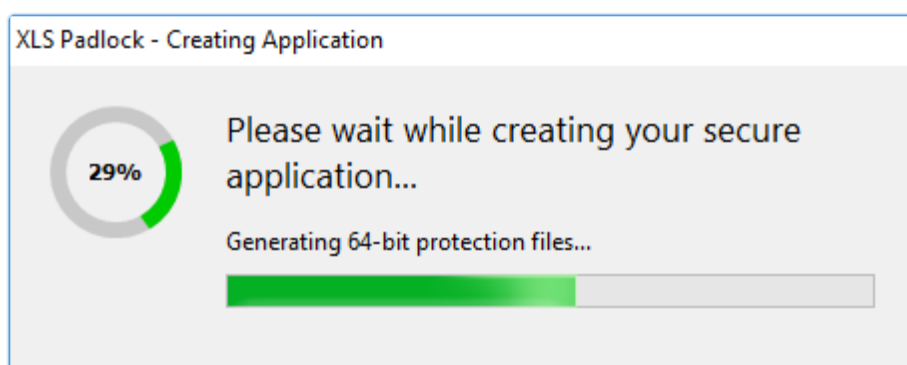
Distribuir uma planilha protegida

Compile sua planilha do Excel para EXE

No XLS Padlock, ao clicar em “Build Secure Application”



, é exibida a caixa de diálogo de progresso da compilação:



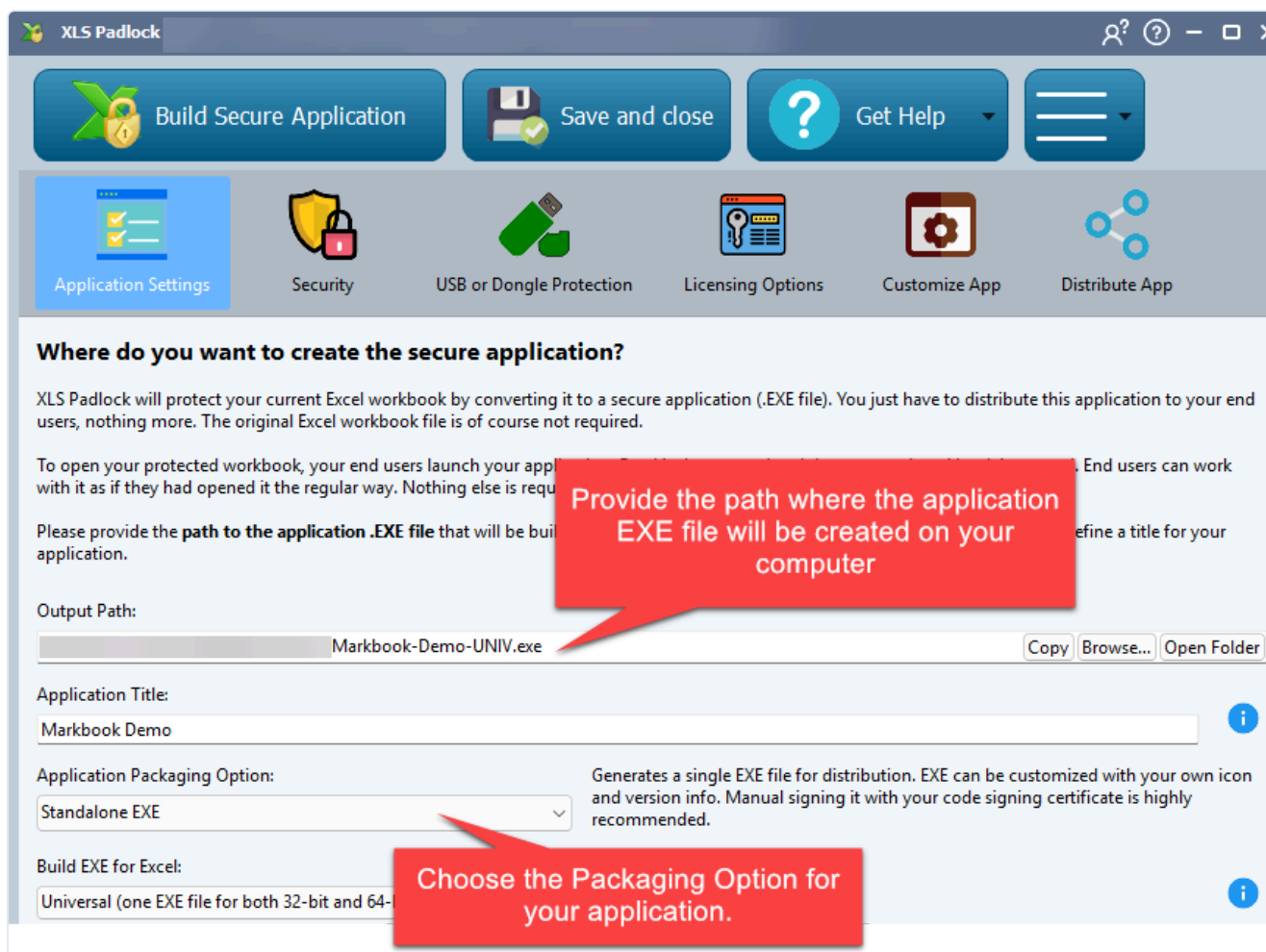
Quando termina, **sua aplicação segura está pronta**. O XLS Padlock gerou um **arquivo executável** (.EXE) a partir da sua planilha do Excel.

▶ Agora, para acessar a planilha protegida, os usuários finais executam este arquivo .EXE. Uma cópia local do Microsoft Excel é o único requisito adicional. Seu arquivo de planilha original **NÃO É MAIS** necessário.

- Ao executar o arquivo .EXE, o Excel é iniciado e a planilha protegida é aberta. Os usuários finais podem interagir com ela como se a tivessem aberto da maneira habitual.
- Determinados recursos do Excel como “New Workbook”, Open e Save (opcionalmente) são desativados por motivos de segurança.
- Se o Excel não for encontrado no computador do usuário, será exibida uma mensagem de erro. Você pode configurar sua aplicação para [exigir versões específicas do Excel](#).

Opções de empacotamento da aplicação

O XLS Padlock oferece dois formatos para sua aplicação segura, o que influencia como você a distribui e gerencia avisos de segurança como o Windows SmartScreen.



EXE independente

O XLS Padlock cria um único arquivo executável (.EXE) a partir da sua planilha. Você simplesmente distribui este único arquivo aos seus usuários finais. A planilha original do Excel não é necessária.

A assinatura de código é altamente recomendada

Aconselhamos fortemente [assinar digitalmente seus arquivos EXE](#). Distribuir um arquivo EXE sem assinatura provavelmente fará com que o Windows SmartScreen exiba um aviso de "Unknown Application". Embora seu EXE continue sendo executado, uma assinatura digital oferece confiança e uma melhor experiência de usuário.

Pacote de aplicação EXE + XPLAPP

Esta opção gera um arquivo EXE, um arquivo complementar `.bin64` e um arquivo de dados `.xplapp` separado. O EXE principal vem pré-assinado pela nossa empresa, o que ajuda no reconhecimento por parte do Windows SmartScreen e do software antivírus.

- Você deve distribuir os três arquivos juntos (`.exe` , `.bin64` e `.xplapp`).
- O XLS Padlock pode agrupar esses arquivos em um [único arquivo Zip](#) ou um [instalador](#) para facilitar a distribuição.

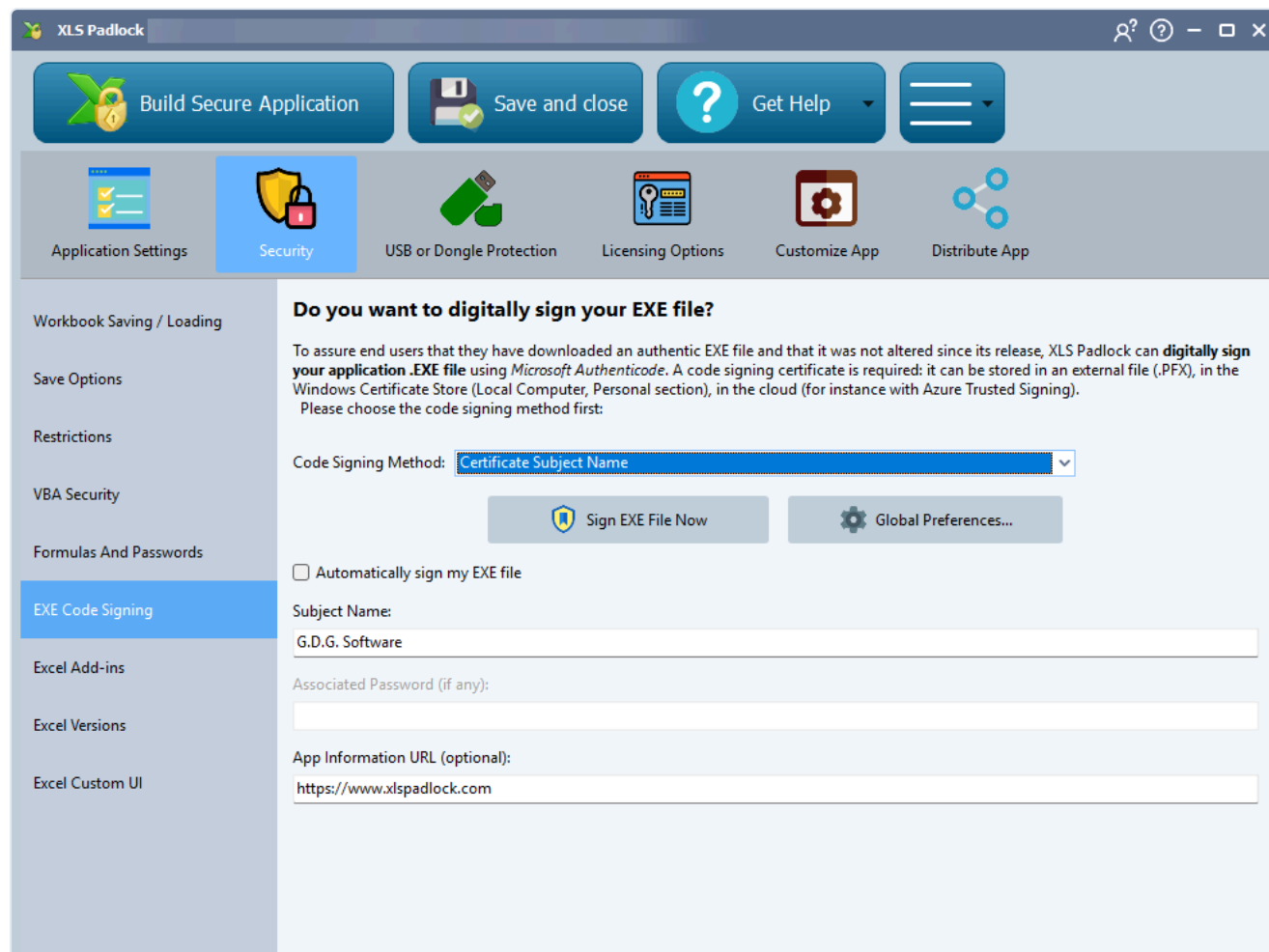
- Esta opção reduz consideravelmente o risco de avisos de "Unknown Application" e de falsos positivos de antivírus.

Recomendações de distribuição

1. **Assine digitalmente seus arquivos EXE:** Este é o passo mais importante para evitar falsos positivos de antivírus e gerar confiança com seus usuários. O XLS Padlock pode [automatizar esse processo](#) se você tiver um certificado de assinatura de código.
2. **Desative temporariamente seu antivírus:** Se você tiver um programa antivírus ativo, considere desativá-lo antes de compilar suas planilhas. O software antivírus às vezes pode marcar incorretamente a criação de novos arquivos EXE no seu computador.
3. **Não envie arquivos EXE por e-mail:** A maioria dos provedores de e-mail bloqueia arquivos `.exe`. Em vez disso, envie os arquivos da sua aplicação para um serviço de hospedagem de arquivos como Dropbox, Google Drive ou seu próprio servidor web, e compartilhe o link de download. Você também pode [empacotar sua aplicação em um arquivo Zip](#).

Assinar digitalmente o EXE

Quando você **assina digitalmente o arquivo EXE da sua planilha independente**, garante aos usuários finais que o arquivo é autêntico e não foi adulterado. Esse processo, também conhecido como assinatura de código, utiliza a tecnologia Microsoft Authenticode® para verificar que o código provém de um editor confiável.



O XLS Padlock simplifica o processo de assinatura de código tratando internamente as etapas necessárias.

Por que a assinatura de código é importante

Se você pretende distribuir as suas aplicações pela Internet, a assinatura de código é altamente recomendada. Ela ajuda a evitar que os navegadores web e o Windows exibam avisos de "Unidentified Publisher" (Editor não identificado) e também pode reduzir a probabilidade de falsos positivos do software antivírus.

![Aviso de segurança do Windows mostrando um Unidentified Publisher para um arquivo EXE não assinado](file:///D:/websites/xlspadlock/astro/src/content/docs/pt/doc/code-sign-your-exe-file-digital-signature/images/image_97.png)

Como obter um certificado de assinatura de código

Para assinar a sua aplicação, você precisa de um **Code Signing Certificate** (certificado de assinatura de código) válido de uma autoridade de certificação (CA) confiável, como Sectigo ou Digicert. Outros tipos de certificado, como SSL/TLS, não são compatíveis.

Certificados baseados em nuvem

Uma alternativa cada vez mais popular e econômica é o **Azure Trusted Signing**, um serviço baseado em nuvem da Microsoft. Ele elimina a necessidade de tokens USB físicos e agiliza o processo de gerenciamento. O XLS Padlock oferece suporte completo à [assinatura com o Azure Trusted Signing](#).

Certificados baseados em token (HSMs)

Desde 1 de junho de 2023, todas as chaves privadas de novos certificados de assinatura de código devem ser armazenadas em hardware seguro, como um token USB compatível com FIPS 140-2 Level 2 ou um Hardware Security Module (HSM). Isso reforça a segurança ao evitar o roubo de chaves. O XLS Padlock funciona perfeitamente com certificados baseados em token; basta garantir que o token esteja conectado ao seu computador quando você compilar a sua aplicação.

Configurando a assinatura de código no XLS Padlock

Na interface do XLS Padlock, vá até a aba **Security -> EXE Code Signing**. Para habilitar a assinatura, escolha o seu **Code Signing Method** (método de assinatura de código) preferido:

- **PFX File:** Utiliza um certificado armazenado em um arquivo `.pfx`. Esse é um método legado para certificados antigos.
- **Certificate Subject Name:** Localiza o certificado no repositório de certificados do Windows pelo seu nome de sujeito (Subject Name). Esse é um método comum para certificados em tokens de hardware.
- **Certificate Thumbprint:** Localiza o certificado no repositório de certificados do Windows pela sua impressão digital única (Thumbprint, um hash SHA-1). Costuma ser o método mais confiável.
- **SignTool Commands:** Um método avançado que permite fornecer comandos personalizados para o utilitário `SignTool.exe` da Microsoft, oferecendo o máximo de flexibilidade.
- **Azure Trusted Signing:** Assina a sua aplicação com o serviço baseado em nuvem da Microsoft. Consulte o nosso [Tutorial do Azure Trusted Signing](#) para mais detalhes.

Azure CLI Required

Para usar esse método, você deve primeiro instalar a CLI do Microsoft Azure e fazer login usando `az login`.

Realizando a assinatura

- **Assinatura manual:** Clique em **Sign EXE File Now** para assinar imediatamente o último arquivo EXE compilado.
- **Assinatura automática:** Marque **Automatically sign my EXE file** para que o XLS Padlock assine o EXE toda vez que você compilar a sua aplicação.

Resolução de problemas

Se ocorrer um erro de assinatura, verifique as mensagens detalhadas no registro de compilação do XLS Padlock. Esse arquivo de registro normalmente recebe o nome [Your Workbook Filename].xplcompil.log e está localizado no mesmo diretório que a sua planilha.

Criar um instalador

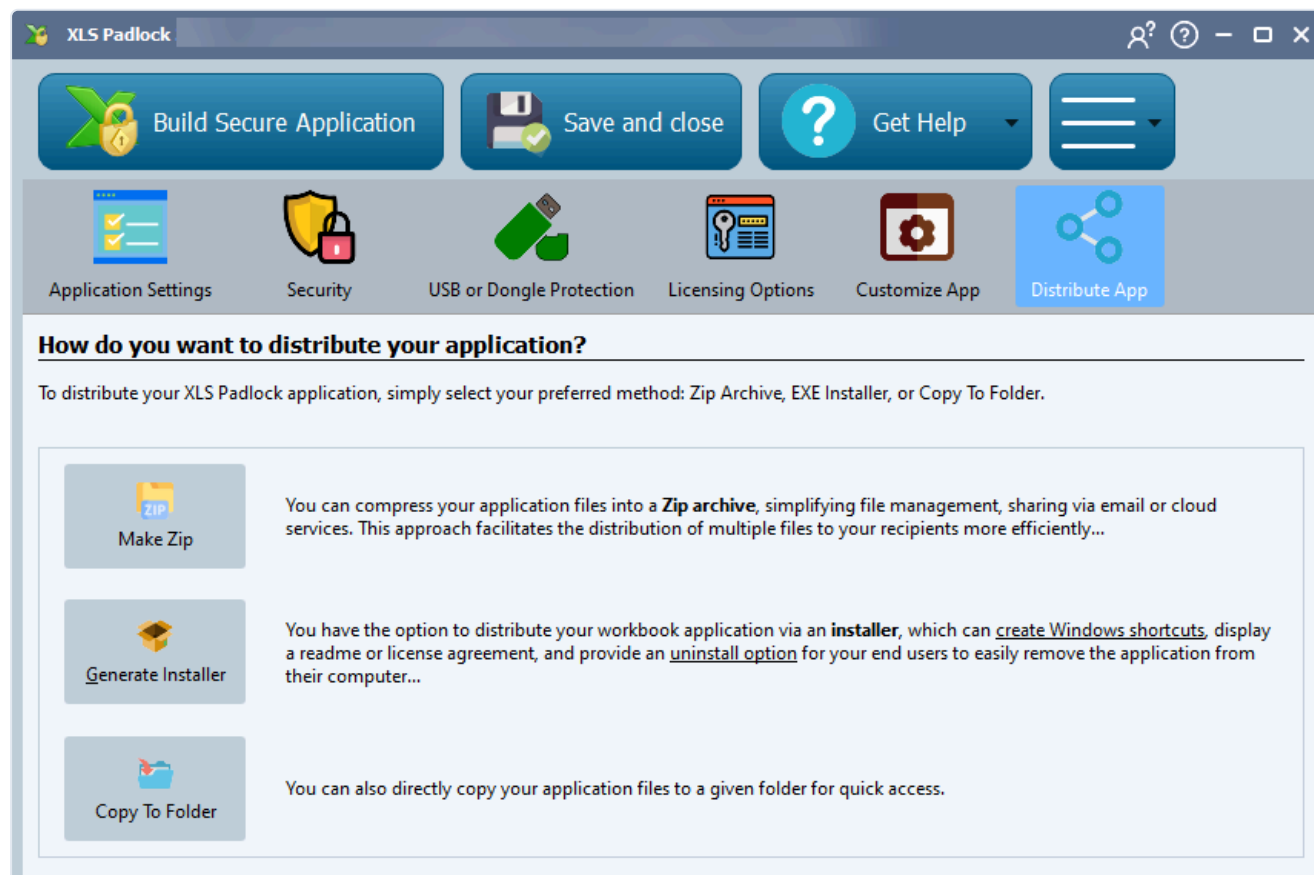
Empacotar a sua planilha compilada em um instalador profissional (ou programa de instalação) e uma excelente forma de distribuir o seu aplicativo. Os instaladores oferecem uma experiência de usuário familiar e podem cuidar de tarefas como instalar vários arquivos (por exemplo, um arquivo leia-me), exibir um contrato de licença, criar atalhos e fornecer um desinstalador para uma remoção limpa.

O XLS Padlock se integra com o nosso software, o [Paquet Builder](#), para gerar instaladores personalizados e compactos. **O Paquet Builder precisa estar instalado no seu computador para usar este recurso.**

O Paquet Builder é uma ferramenta poderosa que combina um criador de arquivos autoextraíveis 7-Zip com um gerador de rotinas de instalação. Ele permite criar instaladores flexíveis e compactos para a distribuição profissional de software, empacotar documentos e arquivos de programa, criar pacotes de instalação multilíngues, gerar atualizações e agrupar vários arquivos em um único `.exe` para facilitar a distribuição on-line.

Para criar um instalador:

1. Clique em **Generate Installer** (gerar instalador) na aba "Distribute" do XLS Padlock. Isso abrirá a janela "Make Setup for your compiled workbook".



1. Preencha o "Destination path" (pasta de instalação padrão), o "Setup title" (título da janela do instalador) e o "Your Application Name" (para atalhos, desinstalador, etc.).

2. Pressione **Generate Installer** para criar o projeto do Paquet Builder. O XLS Padlock então iniciará o Paquet Builder, permitindo que você personalize ainda mais o projeto e compile o 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](#).

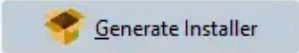
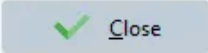
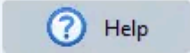
Please fill in the following settings and click **Generate Installer**. A **default project file** to be compiled with Paquet Builder will be created: simply open it in Paquet Builder, modify any settings if needed, and compile it.

Destination Path: c:\program files\
My Company\My Application

Installer Title:
My Application Installer

Your Application Name (used for shortcuts and uninstall display name):
My Application

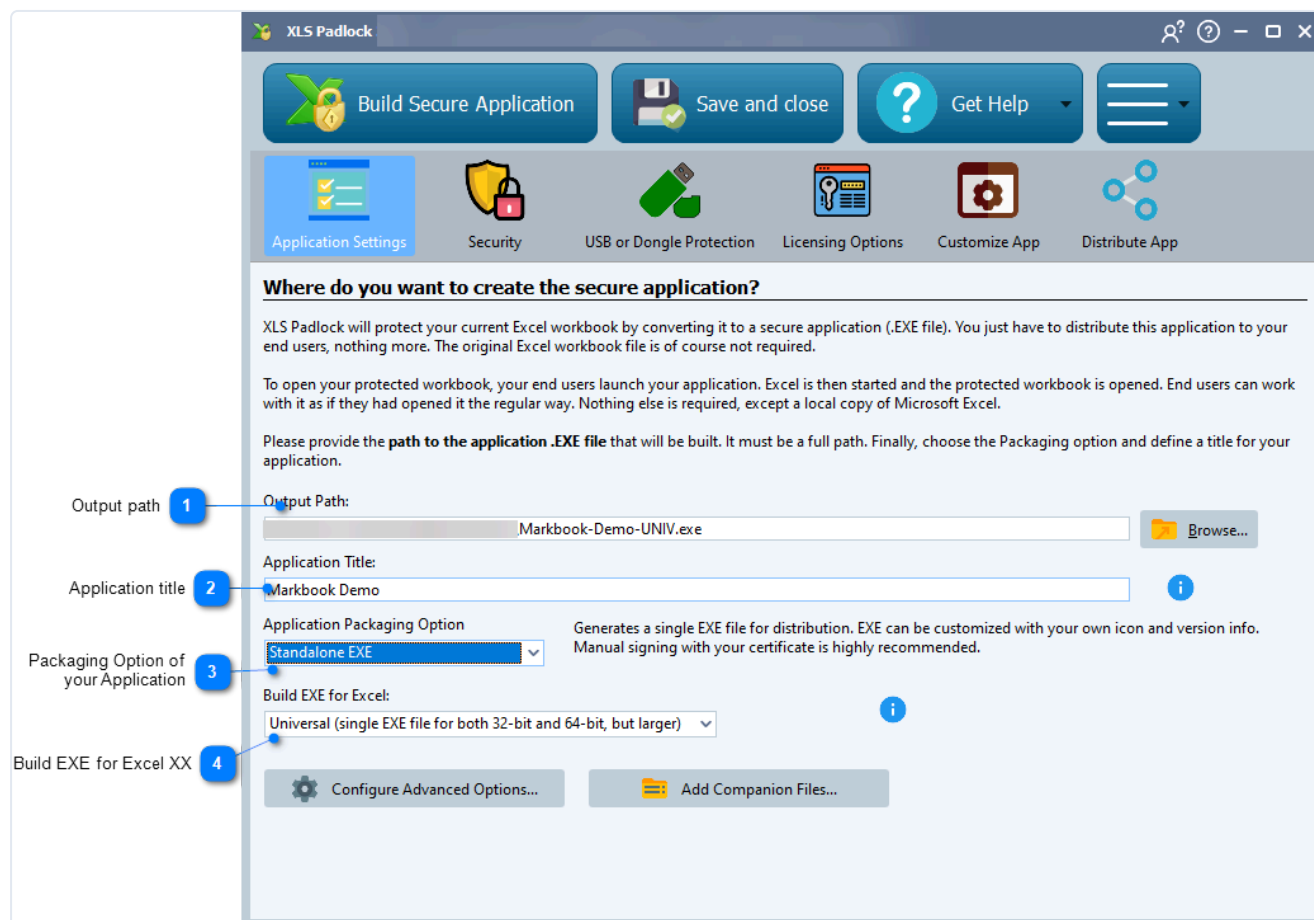
Open the generated project in Paquet Builder

👉 Consulte o [tutorial completo sobre como criar um instalador para a sua planilha do Excel](#).

Compilar o EXE conforme a arquitetura do Excel

O Microsoft Excel está disponível em duas arquiteturas: 32 bits e 64 bits. Como 64 bits é agora o padrão para as versões modernas do Office, a maioria dos seus usuários precisará de uma aplicação compatível com 64 bits. No entanto, alguns podem continuar usando a versão de 32 bits, especialmente em sistemas mais antigos. O XLS Padlock permite compilar para ambas as plataformas.



Esta opção está disponível somente para [arquivos executáveis independentes](#).

👉 Escolha o tipo de arquivo EXE que o XLS Padlock irá gerar:

- **32-bit only**: gera um arquivo EXE compatível apenas com as versões de 32 bits do Excel.
- **64-bit only**: gera um arquivo EXE compatível apenas com as versões de 64 bits do Excel.
- **Universal**: cria um único arquivo EXE que funciona com as versões de 32 bits e de 64 bits do Excel. Isso simplifica a distribuição, mas o arquivo EXE resultante terá aproximadamente o dobro do tamanho.
- **32-bit and 64-bit**: gera dois arquivos EXE independentes simultaneamente. Para evitar que eles se sobrescrevam, o XLS Padlock adiciona automaticamente os sufixos "32" e "64" aos nomes dos arquivos (por exemplo, `MyApp32.exe` e `MyApp64.exe`).

Incompatibilidade de versão

Se um usuário final executar um EXE com uma arquitetura do Office incompatível (por exemplo, um EXE de 64 bits em uma instalação do Office de 32 bits), ele receberá uma mensagem de erro instruindo-o a entrar em contato com você para obter a versão correta.

Assinatura de código recomendada para o modo Universal

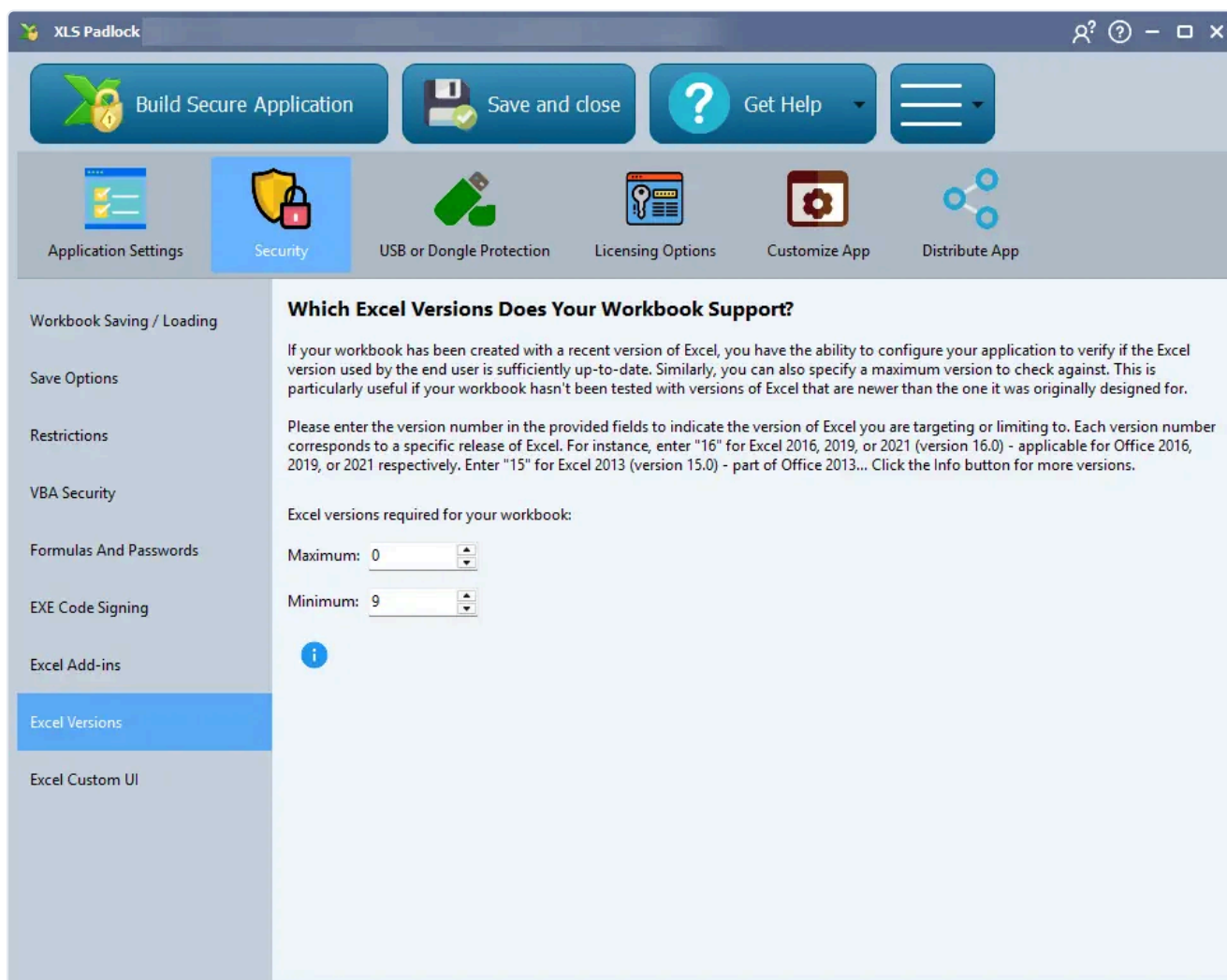
Se você utilizar o modo Universal, recomendamos enfaticamente que assine digitalmente sua aplicação. Isso ocorre porque um arquivo EXE temporário é criado em tempo de execução, o que às vezes pode ser sinalizado pelo software antivírus. Uma assinatura digital reduz consideravelmente o risco de falsos positivos.

Informação

Por padrão, as [aplicações distribuídas no formato de pacote \(bundle\)](#) já são universais.

Versões do Excel

Se sua planilha requer recursos de uma versão específica do Excel, você pode configurar seu aplicativo para verificar a versão do Excel do usuário final em tempo de execução.



Você pode definir tanto uma versão mínima necessária quanto uma versão máxima permitida. Isso é útil para garantir a compatibilidade ou para impedir o uso em versões mais recentes e não testadas do Excel.

Insira o número de versão correspondente nos campos:

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

Atualizações da planilha

Após distribuir seu aplicativo de planilha do Excel, mais cedo ou mais tarde você precisará fornecer atualizações aos seus usuários. O XLS Padlock oferece várias ferramentas para agilizar esse processo.

Uma consideração importante para as atualizações é como seu aplicativo salva os dados do usuário. Você deve conhecer os [diferentes modos de salvamento oferecidos pelo XLS Padlock](#) antes de compilar seu aplicativo.

Se sua planilha for atualizada com frequência e seus usuários só precisarem modificar alguns poucos valores de célula, o modo **Save defined cell values only (.XLSCE file)** (salvar somente os valores de célula definidos) é a melhor escolha. Esse modo permite atualizar o arquivo principal da planilha sem afetar os dados do usuário, pois o XLS Padlock salva e restaura [apenas os valores de célula que você designou](#).

Quando você atualizar sua planilha de origem, a forma mais direta de distribuir a atualização é recompilá-la e enviar o novo arquivo EXE aos seus clientes.

Fluxo de trabalho de atualização recomendado

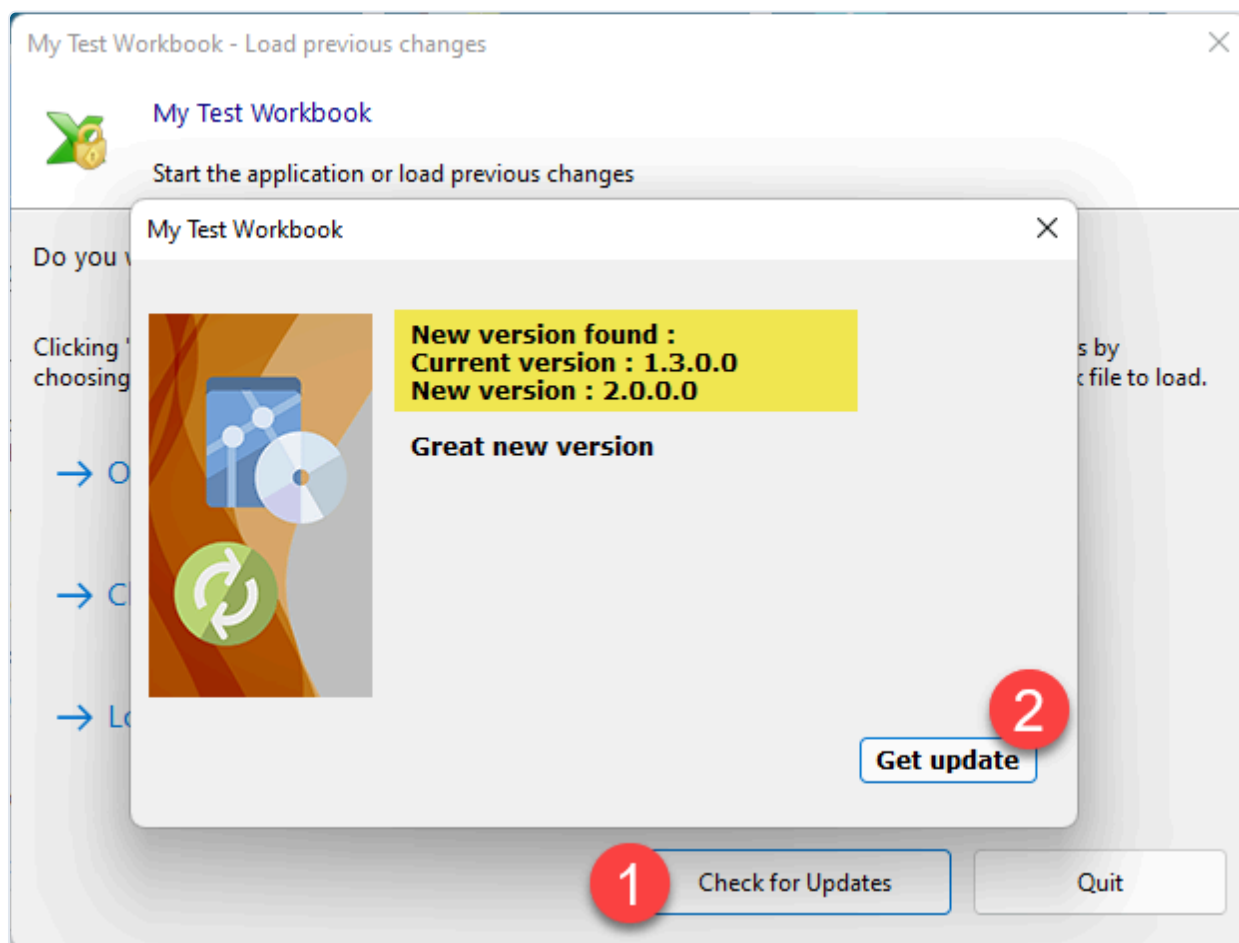
O comportamento no lado do usuário depende do que você alterou entre as versões:

- **Apenas o conteúdo da planilha** (fórmulas, formatação, planilhas, lógica VBA): mantenha o **Application GUID** e a **Secret Key** existentes. No modo `.XLSCE`, os usuários verão sua nova lógica com os valores de célula salvos restaurados. No modo `.XLSC`, os usuários continuarão vendo seu último instantâneo salvo, suas alterações só aparecerão se eles escolherem manualmente "Original Workbook" na tela de boas-vindas.
- **Você quer invalidar completamente os arquivos de salvamento antigos do usuário**: gere uma nova **Secret Key** nas configurações de [Activation and Licensing](#), mas deixe o **Application GUID** inalterado para que as chaves de ativação existentes permaneçam válidas. Na primeira inicialização do usuário, o arquivo de salvamento antigo será detectado como incompatível e ele será solicitado, por meio da tela de boas-vindas, a carregar em vez disso a planilha original incorporada.
- **Não altere entre modos de salvamento entre versões** (`.XLSC` ↔ `.XLSCE`). Os arquivos de salvamento existentes do modo anterior serão ignorados, e todos os dados que seus usuários ainda não tiverem reinserido serão perdidos. Escolha um modo de salvamento no início do projeto e mantenha-o durante toda a vida útil do aplicativo.

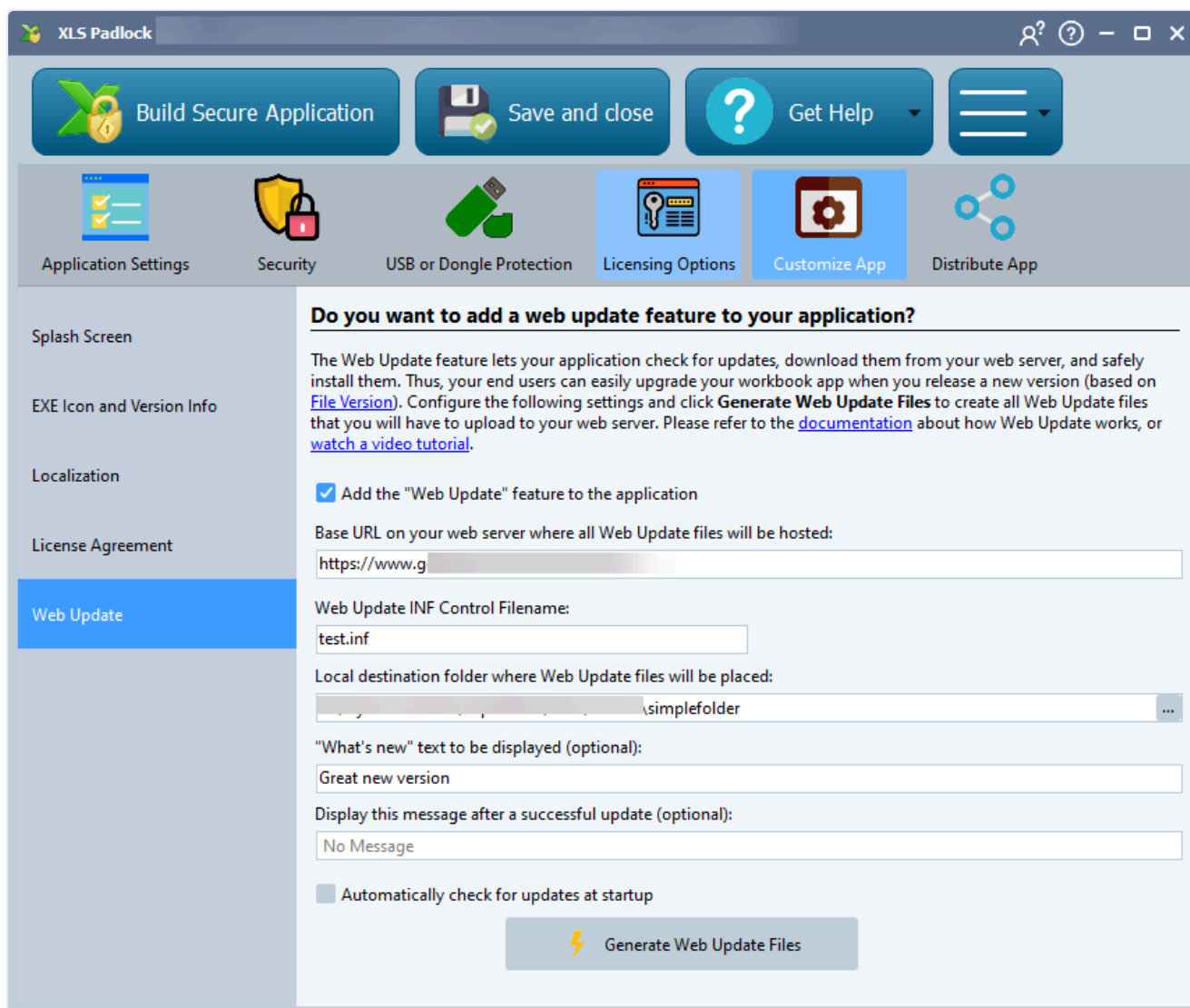
DICA

O botão [Original Workbook na tela de boas-vindas](#) funciona como rede de segurança do usuário quando um arquivo de salvamento antigo não consegue ser carregado. Mesmo que você o tenha ocultado, o XLS Padlock o reativará automaticamente para a tentativa de repetição, de modo que os usuários nunca fiquem bloqueados.

👉 No entanto, baixar e substituir manualmente o arquivo EXE pode ser tedioso para os usuários finais. Para simplificar isso, o XLS Padlock oferece um recurso de atualização web automática. Você pode configurar seu aplicativo para verificar se há atualizações on-line e notificar os usuários quando uma nova versão estiver disponível:



Esse recurso pode ser configurado de forma rápida e diretamente no XLS Padlock:



👉 Para configurar seu próprio sistema de atualização, você precisará de um servidor web ou de um espaço de hospedagem onde possa colocar arquivos para download direto. Em seguida, siga as instruções da página [como configurar atualizações web automáticas](#).

Tutorial em vídeo

Assista ao nosso tutorial em vídeo sobre [como configurar atualizações web automatizadas para suas planilhas do Excel](#).

Atualizações web automáticas

O XLS Padlock inclui um recurso de atualização web que pode baixar e instalar automaticamente novas versões da sua aplicação. Isso permite que seus usuários finais atualizem facilmente o aplicativo da sua planilha sempre que você lançar uma nova versão.

Nenhum software de terceiros é necessário, mas **você precisa ter um servidor web ou um espaço de hospedagem web** para hospedar os arquivos de atualização.

Tutorial em vídeo

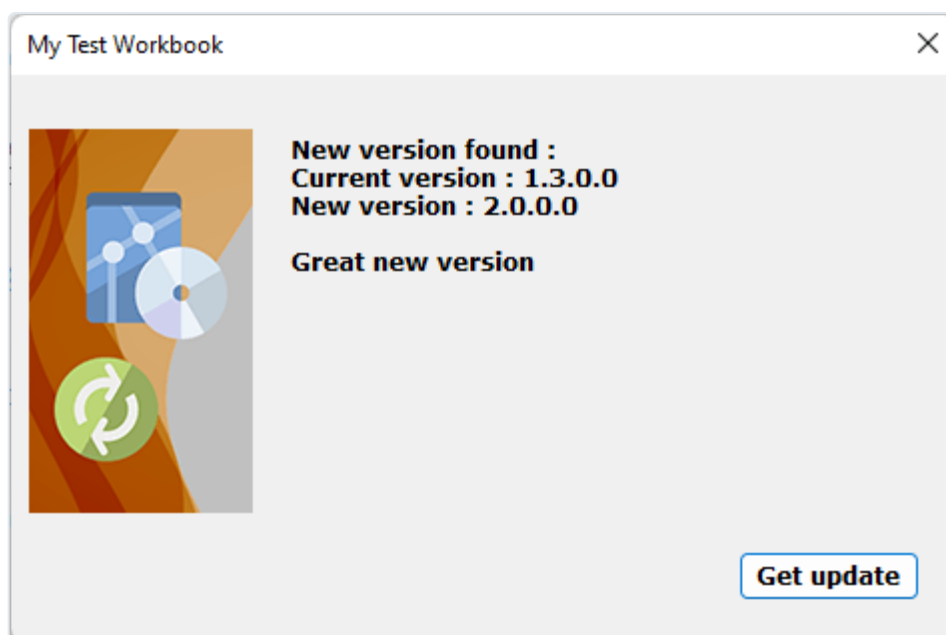
Assista a um tutorial em vídeo sobre como configurar atualizações web automáticas para planilhas do Excel.

Como funciona a atualização web

A aplicação baixa primeiro um pequeno arquivo de controle (`.inf`) do seu servidor. Esse arquivo informa à aplicação se há uma nova versão disponível. Se houver, a aplicação baixa os arquivos de atualização necessários (`.cab`), fecha-se, instala a atualização e reinicia.

Verificar atualizações

Você pode configurar sua aplicação para **verificar automaticamente atualizações na inicialização** ou adicionar um botão "**Check for Updates**" (verificar atualizações) à [tela de boas-vindas](#). Quando uma nova versão é detectada, os usuários são guiados por um assistente de atualização web.



Os usuários também podem iniciar uma atualização por meio do [parâmetro de linha de comando](#) `-webupdate`.

Como uma nova versão é detectada

O número de **File Version** (versão do arquivo), especificado na página [EXE Version Info](#) do XLS Padlock, é usado para determinar se uma atualização é necessária. Você deve aumentar esse número de versão toda vez que lançar uma atualização.

Configuração

- **Base URL:** A URL no seu servidor web onde os arquivos de atualização web serão hospedados (por exemplo, `https://www.yourwebsite.com/myfolder`). HTTPS é recomendado.
- **Web Update INF Control Filename:** O nome do arquivo de controle que contém as informações de atualização.
- **Local Destination Folder:** Uma pasta local no seu computador onde o XLS Padlock gerará os arquivos de atualização. Em seguida, você deve enviar todo o conteúdo dessa pasta para o seu servidor web.

Aviso

A pasta de destino deve estar vazia. O XLS Padlock pedirá para excluir o conteúdo dela caso não esteja.

- **"What's new" text (opcional):** Texto a ser exibido no assistente de atualização, informando aos usuários sobre as mudanças na nova versão.

Gerar os arquivos de atualização web

Clicar em **Generate Web Update Files** no XLS Padlock criará os arquivos necessários (um arquivo de controle `.inf` e um arquivo compactado `.cab`) na pasta local que você especificou. Em seguida, você pode enviar esses arquivos para o seu servidor web.

Privilegios de administrador

Se a aplicação estiver instalada em uma pasta restrita do Windows, como `Program Files`, serão necessários privilégios de administrador para instalar a atualização, o que acionará um aviso do UAC.

Resolução de problemas

Para resolver problemas com o recurso de atualização web, ative a opção **"Enable WebUpdate Log"** (habilitar log de atualização web) em [Opções avançadas](#). Isso criará um arquivo `WUPDATE.LOG` no diretório Documentos do usuário com informações detalhadas sobre o processo de atualização.

Migrar os dados do usuário entre atualizações

Ao atualizar sua planilha protegida e distribuir um novo EXE, você pode enfrentar dificuldades para garantir que os dados de seus usuários sejam transferidos corretamente, especialmente se você usar o **modo Full Save**. Esse modo cria um **instantâneo completo da planilha** no momento de salvar. Por consequência, quando um usuário abre seu antigo arquivo salvo (`.xls`) com o novo aplicativo, ele verá seus dados antigos na antiga estrutura da planilha e não se beneficiará de suas últimas atualizações (como novos recursos ou correções de erros em seu código VBA).

Este tópico explica como usar um sistema de exportação/importação baseado em VBA para ajudar seus usuários a migrar seus dados de uma versão antiga de seu aplicativo para uma nova.

Aviso

A solução descrita aqui depende do **modo de salvamento que você selecionou para seu projeto do XLS Padlock**. Se você usar o modo de salvamento **Cell Values**, os dados do usuário são carregados automaticamente na estrutura de sua planilha atualizada, e este processo manual não é necessário.

Alternativa: modo de salvamento Cell Values

Como mencionado acima, o **modo de salvamento Cell Values** é a forma mais simples de lidar com as atualizações. No entanto, esse modo foi projetado para salvar apenas os valores de células específicas. Se você precisar transferir outros tipos de dados, como o estado de variáveis ou objetos do VBA, será necessário usar uma solução personalizada de VBA como a descrita neste tópico.

Exemplo de código

Esse processo envolve duas etapas principais: exportar os dados da versão antiga e importá-los para a nova.

1. Exportar os dados do usuário para um arquivo do Excel

A macro abaixo gerará um arquivo do Excel `.xlsx` normal contendo os dados do usuário. Você deve incluir essa macro em seu antigo aplicativo, vinculada a um botão, para que os usuários possam exportar seus dados antes de atualizar.

OBSERVAÇÃO

Você precisará substituir os nomes das planilhas e os intervalos de células na macro abaixo para que correspondam à estrutura da sua planilha.

```
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("A1:C3").Value  
    ActiveWorkbook.Sheets("SheetB").Range("B3").Value = ThisWorkbook.Sheets("SheetB").Range("B3").Value  
    ActiveWorkbook.Sheets("SheetC").Range("B1:C3").Value = ThisWorkbook.Sheets("SheetC").Range("B1:C3").Value  
    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. Carregar os dados no novo arquivo EXE.

O usuário precisa abrir o novo EXE e executar a 3ª macro abaixo para carregar os dados (vincule a macro a um botão). Assim que o usuário executar a macro, será solicitado que ele selecione o arquivo (1ª macro), e os dados serão copiados (2ª macro). A terceira macro executará ambas as macros, e essa é a macro que precisa ser vinculada ao botão. Novamente, você precisará alterar os nomes das células e das abas na segunda macro, e também pode alterar o título na primeira 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
```

OBSERVAÇÃO

Para salvar planilhas normais, você também pode consultar: [Carregar/salvar planilhas por meio do assistente VBA SetOption](#)

👉 Veja também:

- [Modo de salvamento que você selecionou para seu projeto do XLS Padlock](#)

Salvar e restaurar as configurações por meio de modelos

O XLS Padlock permite que você salve as configurações atuais do seu projeto em um arquivo de modelo. Este é um recurso poderoso que pode economizar uma quantidade significativa de tempo, especialmente se você cria com frequência novas aplicações de planilha com configurações semelhantes.

Salvar as configurações do projeto em um modelo

Depois de configurar todas as opções desejadas para o seu projeto (como configurações de segurança, personalização e opções de saída), você pode salvar essa configuração como um modelo.

Clique no botão



App Menu (menu do aplicativo) e, em seguida, no menu **Save Current Settings As Template**. Será solicitado que você escolha um local e um nome para o seu arquivo de modelo (que terá a extensão `.XPLP`).

Aviso

Não escolha o seu arquivo de projeto existente do XLS Padlock.

Restaurar configurações a partir de um modelo

Quando você inicia um novo projeto ou deseja aplicar uma configuração padrão a um projeto existente, pode restaurar facilmente as configurações a partir de um modelo salvo anteriormente.

Clique no botão



App Menu, depois no menu **Restore Project from Template**, navegue até o seu arquivo de modelo `.XPLP` e selecione-o. Todas as configurações armazenadas nesse modelo serão aplicadas instantaneamente ao seu projeto atual.

OBSERVAÇÃO

O [caminho de saída da aplicação](#) não é armazenado dentro dos modelos.

Erro de registro ou EREGISTRYEXCEPTION

Esse erro geralmente indica um conflito com um software de segurança ou um programa antivírus. Para resolvê-lo, ative a opção **"Use another registry key for storing activation data"** (usar outra chave de registro para armazenar os dados de ativação) nas [Opções avançadas](#).

Violação de acesso no endereço

Esse erro pode ocorrer ao usar o recurso "Lock VBA Project (simple VBA protection)" (bloquear o projeto VBA, proteção VBA simples). Para resolvê-lo, remova a senha do seu projeto VBA antes de compilar a planilha com o XLS Padlock. A senha não é necessária, pois o XLS Padlock bloqueará o projeto automaticamente, sem solicitar nenhuma.

Como alternativa, você pode tentar usar uma senha mais curta ou mudar para a opção [Prevent access to VBA editor](#).

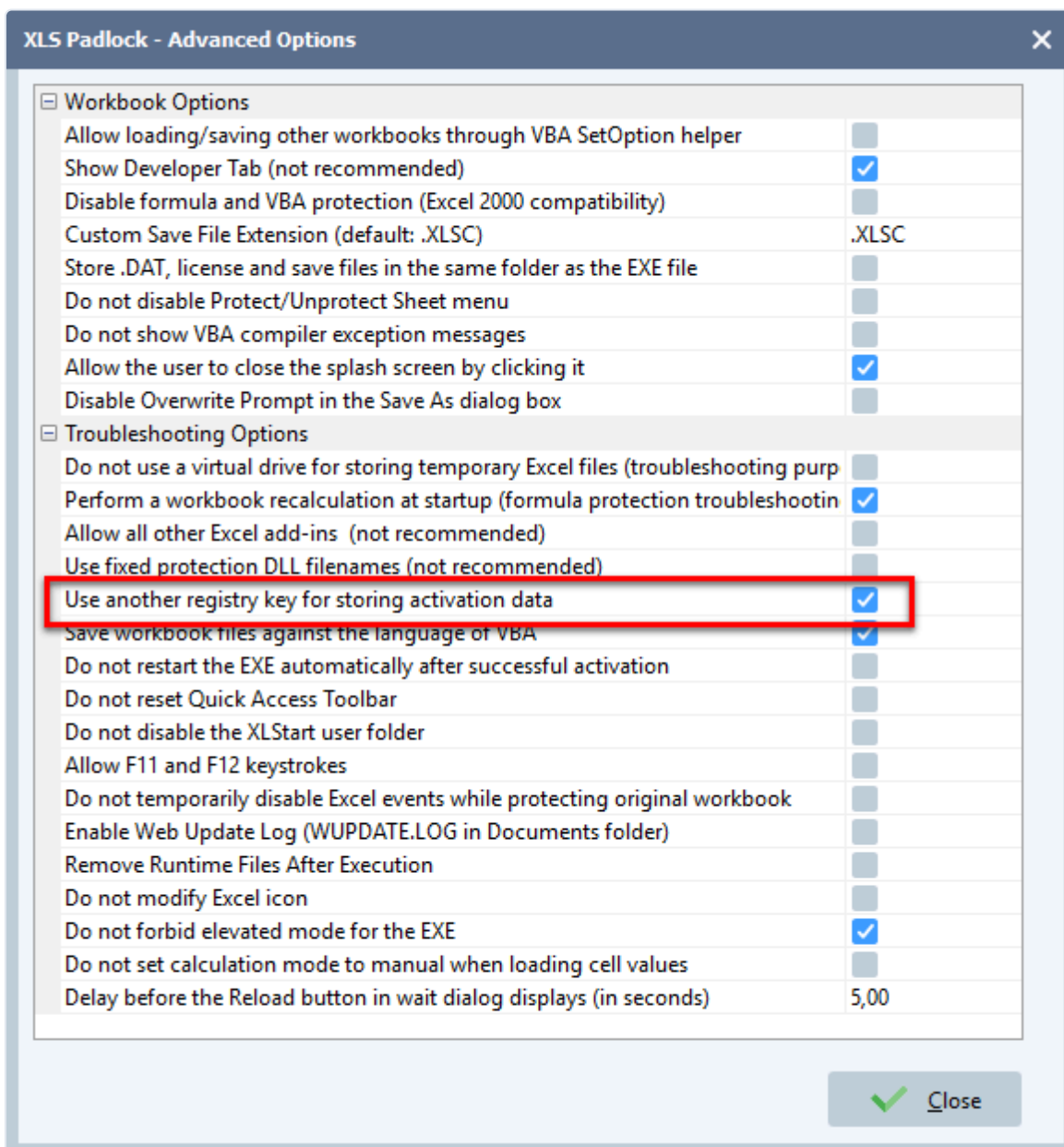
Erro "Failed to set data for 'Data'"

Se o seu usuário final receber a seguinte mensagem de erro ao inserir uma chave de ativação, isso normalmente indica um conflito com um software de segurança em seu sistema.

Error setting data in registry, Failed to set data for 'Data'

Esse erro ocorre porque outro programa, normalmente um antivírus ou uma suíte de segurança, impede que a aplicação grave seus dados de ativação no Registro do Windows.

Para resolver isso, você pode configurar o seu projeto do XLS Padlock para usar uma chave de registro alternativa. Habilite a opção avançada **Use another registry key for storing activation data** (Usar outra chave de registro para armazenar os dados de ativação), conforme mostrado abaixo, e recompile a sua aplicação.



Por que o EXE é tão grande

O arquivo `.exe` compilado é consideravelmente maior do que a planilha original porque é uma aplicação autônoma que agrupa vários componentes essenciais:

- Um carregador seguro compatível com diversas versões do Excel e do Windows.
- Um motor de execução completo com suporte a Unicode.
- O compilador e interpretador VBA proprietário.
- Múltiplas camadas de proteção contra pirataria e de proteção de código.

Para reduzir o tamanho do arquivo EXE:

- **Use UPX Compression** (usar a compactação UPX): UPX é um compactador de arquivos EXE gratuito e de código aberto, disponível em upx.github.io. Compactar o EXE pode reduzir significativamente o seu tamanho. No entanto, recomendamos fortemente assinar digitalmente qualquer arquivo EXE compactado, pois, caso contrário, alguns programas antivírus podem gerar falsos positivos.
- **Disable Formula Protection** (desativar a proteção de fórmulas): se a sua aplicação não depender de fórmulas sensíveis, você pode desativar a proteção de fórmulas para reduzir ligeiramente o tamanho final do arquivo EXE.

Restaurar o arquivo XLS a partir do EXE

Não, não é possível restaurar o arquivo Excel original a partir de um .EXE compilado se o recurso "Save As" (Salvar Como) estava desativado.

Se você precisa da capacidade de recuperar a planilha original, é necessário ativar o recurso "Save As" durante a compilação. Isso permite que você utilize a funcionalidade de [descriptografar arquivos salvos](#).

Links de suporte

Se tiver alguma dúvida ou problema com o XLS Padlock, você pode entrar em contato conosco pelos seguintes canais:

- **Fórum de usuários:** <https://www.gdgsoft.info> (Escolha a categoria "XLS Padlock")
- **Suporte por e-mail:** info@xlspadlock.com
- **Para suporte de Enky SL/LC Dongle:** info@hs-securityware.com

Ao solicitar suporte, inclua a sua versão do Excel (por exemplo, Excel 365, Excel 2019) e a arquitetura (32 bits ou 64 bits) para nos ajudar a resolver o seu problema mais rápido.

Siga-nos no X: <https://x.com/gdgsoft>

Nossos outros produtos estão disponíveis em: <https://www.gdgsoft.com>