

# Unviewable+ VBA for Excel, PowerPoint & Word

CONFIDENTIAL - FOR CLIENT RELEASE ONLY – 35 PAGES

Last updated: 25 February 2019

## INTRODUCTION

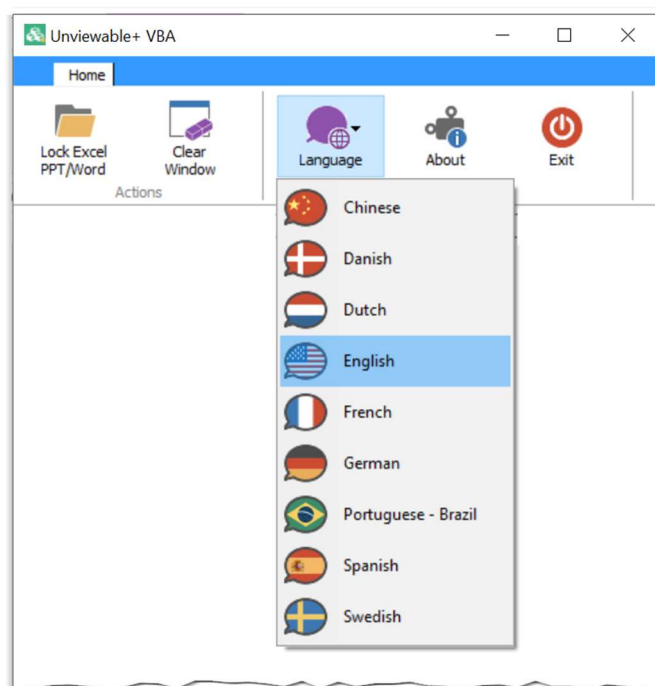
The **UnviewablePlus-vXXX.exe** is a portable Delphi application. It is not an Office add-in. The EXE file is the **windows application** that will make macro-enabled Office Excel, PowerPoint or Word files unviewable in Visual Basic Editor (VBE).

The software will not install anything on your computer. The application does not travel with Microsoft Office documents. It is used for locking Office files only.

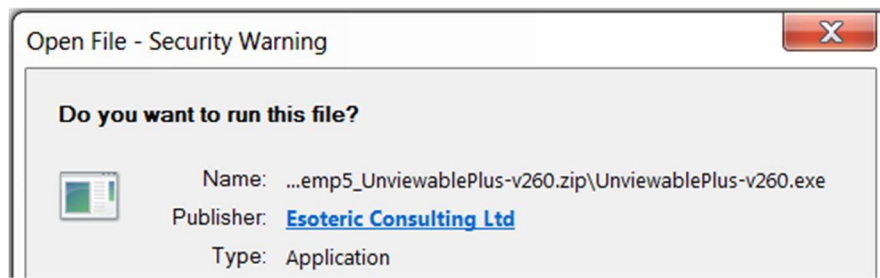
The executable can be renamed as **UnviewablePlus.exe** and copied to your desktop or Excel's add-in folder for use with our companion **VBA Tools** add-in, which will scan both the EXE current folder and the desktop to launch UnviewablePlus.exe

Version agnostic addin folder: %appdata%\Microsoft\AddIns

The **.lng** language files are optional and can be copied to the EXE current folder or to any subfolder below the Delphi executable, if a tidier file layout is preferred.



The executable is signed by a digital certificate from version 2.60 beta onwards.



For the latest documentation updates, please visit the Unviewable+ web page:  
<http://www.spreadsheet1.com/unviewable-vba-project-app-for-excel.html>

Help us improve our software or to report a problem, please contact us:  
<http://www.spreadsheet1.com/contact-us.html> or [petros@spreadsheet1.com](mailto:petros@spreadsheet1.com)

Software updates are free for perpetual license holders. You will be notified by email when a new application release or revised material is available for download from the permanent links in this publication.

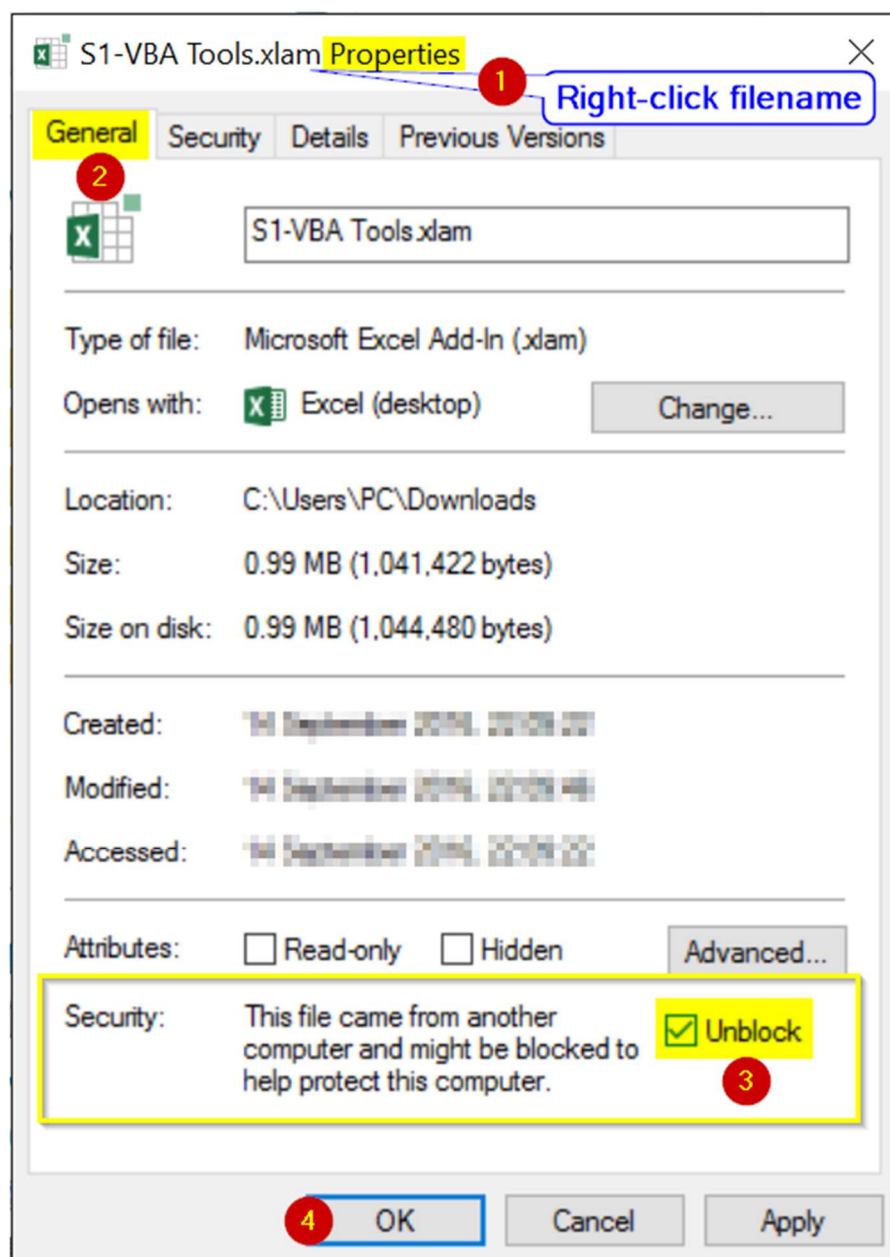
## VBA TOOLS add-in fails to load in Excel 2010 or later

Microsoft released an update in **July 2016** to block Office files and add-ins downloaded from the internet from loading automatically. The feature is called Protected View. Thus, you may not be able to view the ribbon user interface of the VBA Tools add-in, after installing in an add-in folder or opening it.

[1] To unblock the add-in, right-click on its filename and click **Properties**

[2] Mark the **Unblock** checkbox near the bottom of the form and click OK

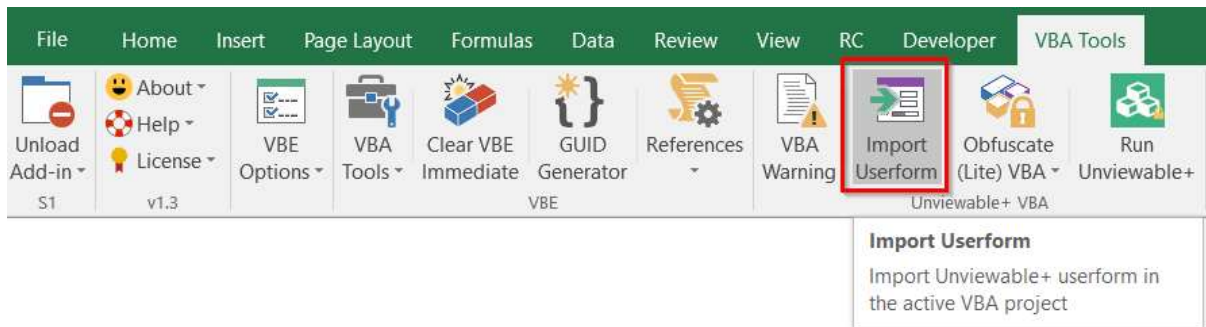
[3] Open the VBA addin or copy it in the Excel add-in folder, so it appears in the list under [ Developer Tab > Excel Add-ins ]



# HOW TO PREPARE VBA PROJECTS FOR LOCKING

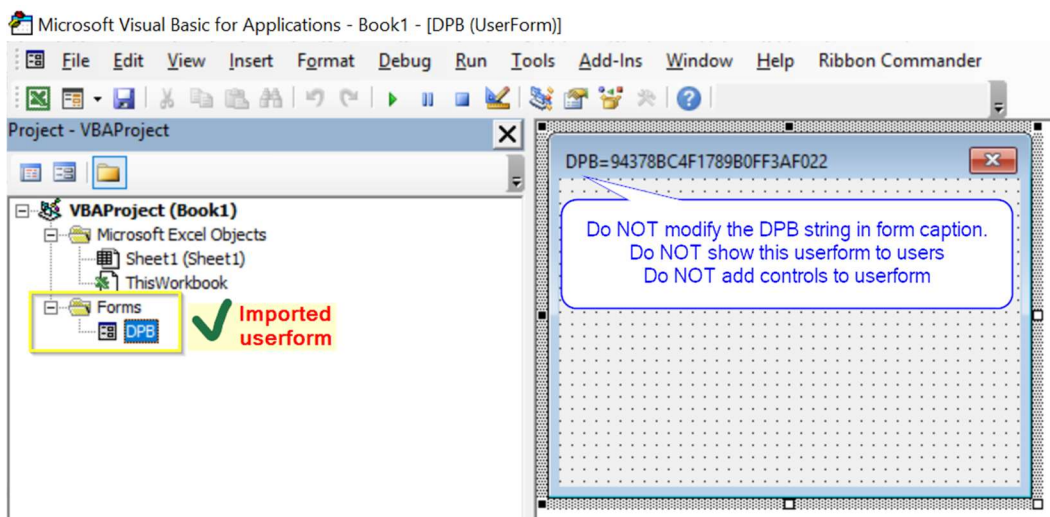
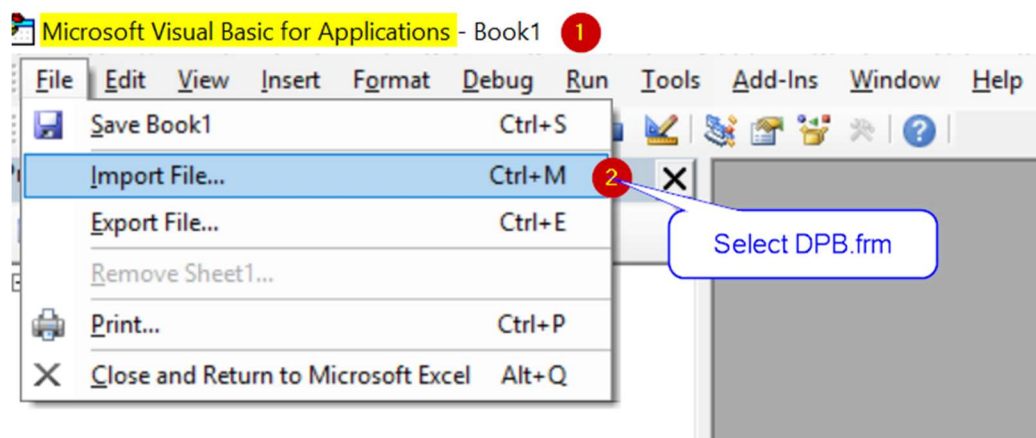
## [1] Use the DPB Userform

Please import the userform **DPB** in any Office VBA project you would like to protect. There is no VBA code in the userform. [1.1] Use the ribbon button in **VBA Tools**:



[1.2] Download the userform from link below and import it from VBE file menu:

<https://spreadsheet1.box.com/s/jahbbe5doyvhxardzj474egh7p9evccf>

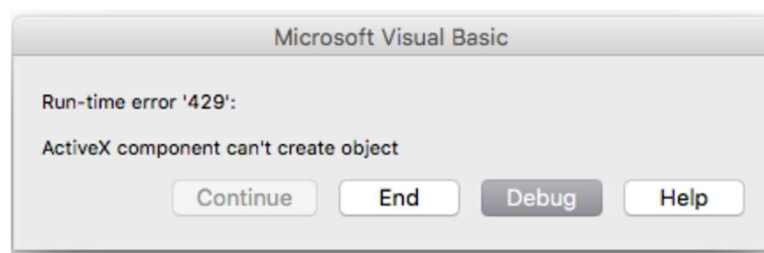


[**Optional**] You can use the following code to detect file tampering. The code should be run at file open (the OnLoad ribbon callback is preferred) and/or when user interface ribbon controls are clicked.

```
If Left$(DPB.Caption, 4) <> "DPB=" Then  
    'Show message or quit app or alert administrator silently  
End If
```

Even if you do not plan to use event code at file open, please import the DPB form without using the VBA detection code shown. **The DPB userform enhances security when imported in VBA projects, even without any VBA code in it.**

- Do not rename for userform and do not show the DPB userform to users
- Do not modify the first 4 characters in the caption. You can modify the hex number using a length between 20-30 characters.
- Use Left\$() instead of Left() to avoid this runtime error in Mac computers:



## [2] Detect VBA Backdooring Applications

It is strongly advisable to close a macro-enabled file immediately upon loading, if it has been opened via a known VBA backdoor application. Due to the nature of the backdoor trick, the Excel file has to be opened from within the hacking software. So the hacking tool is always loaded in memory before an addin or macro-enabled file is opened in Microsoft Office.

```
Private Sub Workbook_Open()
```

```
    If IsBackdoorRunning Then
```

```
        ThisWorkbook.Close SaveChanges:=False
```

```
    Endif
```

```
End Sub
```

---

```
Public Function IsBackdoorRunning() As Boolean
```

```
    Dim objList As Object
```

```
    Set objList = GetObject("winmgmts:").ExecQuery("select * from win32_process where  
Name='VBAPass.exe' or Name='aopr.exe'")
```

```
    IsBackdoorRunning = objList.Count > 0
```

```
End Function
```

The Unviewable+ application cannot insert VBA Code in your VBA projects, as Unviewable+ is a Delphi application and not an Office COM addin. Please edit your projects manually.

Powerpoint and Word do not offer any accessible Open event, so the recommended VBA backdoor detection code should be incorporated in a XML based **OnLoad callback** that is executed on file open, before any other events.

A demo workbook is available in the VBA samples bundled with the software.

### [3] **Caution: Worksheet VBA code can be read, even if VBE is locked**

VBA code stored in worksheets (document modules) can be read from workbooks with unprotected structure, even if the VBA project is password locked or even unviewable! The hack can be demonstrated with the following simple steps:

- Store event code in a worksheet
- Lock the VBA project with a password (VBE>Tools>VBAProject Properties>Protection)
- Copy the worksheet with the event code to a new workbook

The new workbook has its own VBA project, which is unlocked at the time of creation. Obviously, the VBA project password cannot be inherited from the source workbook. Thus, the code embedded in the newly copied worksheet will be visible, even if the source VBA project was password locked or unviewable!

The workaround for this problem involves 2 steps:

- **Protect the workbook's structure to prevent worksheet copying:**  
File tab> Info > Protect Workbook > Protect Workbook Structure or Review > Protect Workbook > Protect workbook for structure
- **Include only minimal code in the worksheets. Place the bulk of any such code in a standard module and call on it from the worksheet**

Tip article: <http://www.spreadsheet1.com/read-document-vba.html>

Sheet & Workbook protection in Excel 2010 or earlier

<http://www.spreadsheet1.com/sheet-protection.html>

Sheet & Workbook protection in Excel 2013 and later

<http://www.spreadsheet1.com/sheet-protection-2013.html>

Excel protection tips:

<http://www.spreadsheet1.com/excel-protection-myths-busted.html>

Printable password log template:

<http://www.spreadsheet1.com/printable-password-log-excel-template.html>

Our password recovery and removal software:

<http://www.spreadsheet1.com/password-protection-recovery-excel-add-in.html>

<http://www.spreadsheet1.com/office-password-eraser.html>

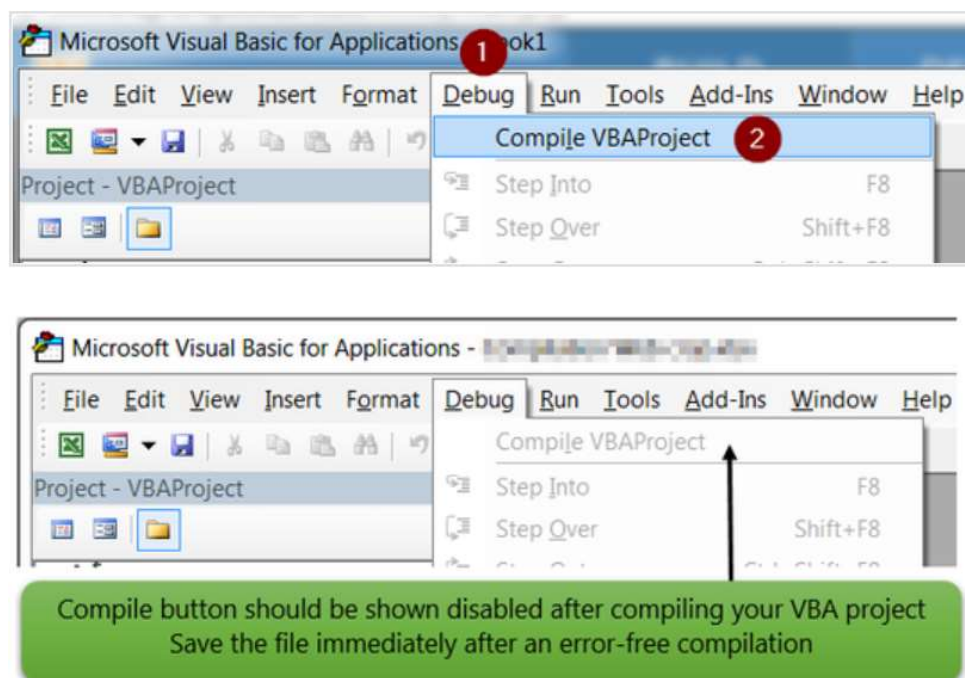
#### [4] Use Option Explicit & Compile VBA Projects

Explicit declaration of variables in all code modules (standard, userform, class) using **Option Explicit** is strongly recommended as a best development practice, but not required. It is not easy to troubleshoot bugs in protected projects, so try to find and correct bugs in the early stages of code development.

Read more: <https://msdn.microsoft.com/en-us/library/office/gg278855.aspx>

Please compile your unlocked VBA source program to verify that is error free, then **save** it.

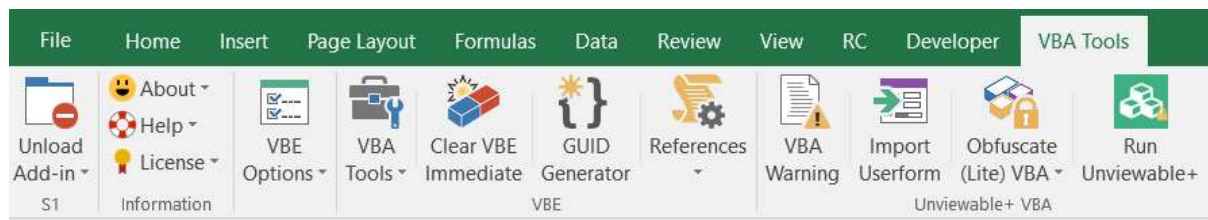
- Please **repeat the compilation process in every Excel & Windows version** that you plan to execute your application, as some VBA properties and methods may not be backwards compatible with older Excel versions.
- In addition, please also ensure that any 3<sup>rd</sup> party libraries referenced are available at user machines.



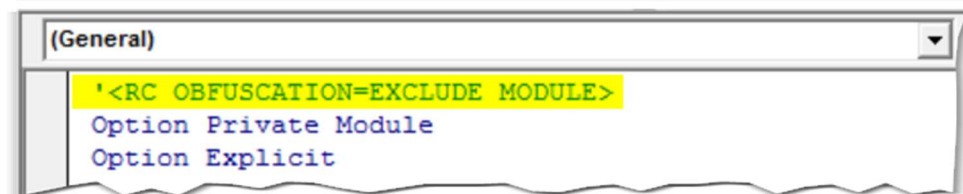
Please ensure that at least one **backup copy of your source code** is stored in at least one safe location, as the original commented VBA code cannot be recovered from obfuscated and locked Office projects.



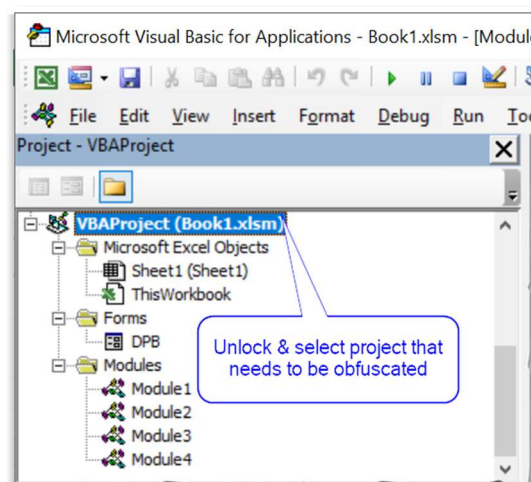
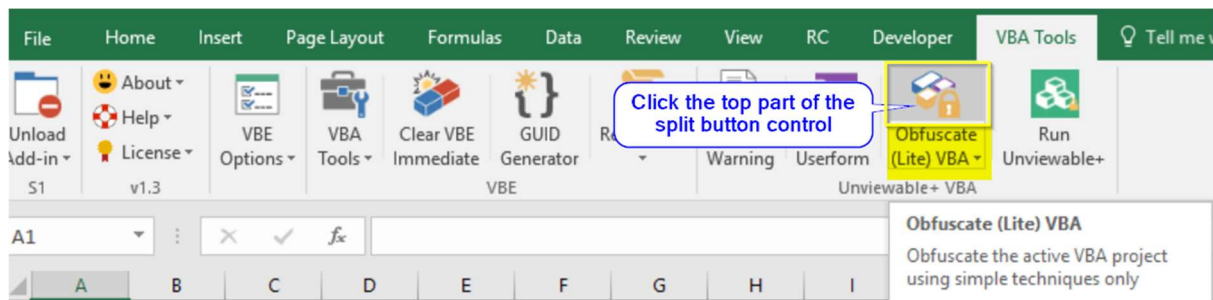
## [5] Obfuscate your VBA code



The companion **VBA Tools** add-in (xlam) offers an elementary code obfuscator that compliments Unviewable+ VBA protection. The program will remove code indentation, blank lines and comments that are redundant inside locked files destined for publication. You can add the following comment in the first line of any code module to prevent it from being obfuscated (e.g. for public domain code for which attribution may be required): '**<RC OBFUSCATION=EXCLUDE MODULE>**'

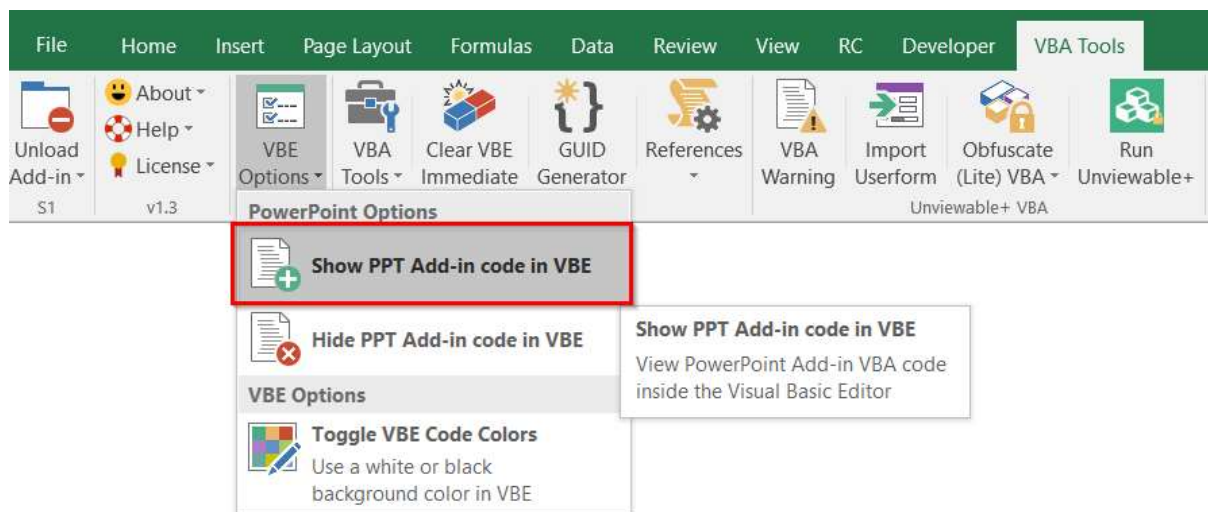


Please ensure that your VBA program is **compiled** before and after obfuscation to verify that your code is error free before being locked.



1. Load **VBA Tools** addin, if it is not already open in Excel
2. Open your VBA file
3. Unprotect your VBA project, if a standard VBE password has been used
4. Ensure your VBA project is selected in VBE
5. Click Obfuscate (Lite) VBA, the top part of control, as it is a split-button (button & menu)
6. Re-compile your VBA project to verify that there are no errors

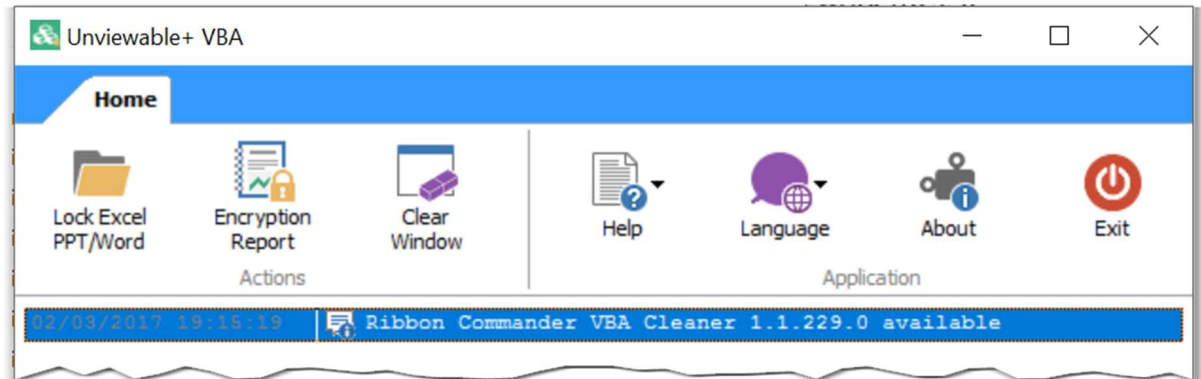
Show/hide add-in VBA code in **PowerPoint**, which is hidden by default, using the VBE Options menu.



Please ensure that at least one **backup copy of your source code** is stored in at least one safe location, as the original commented VBA code cannot be recovered from obfuscated and locked Office projects.

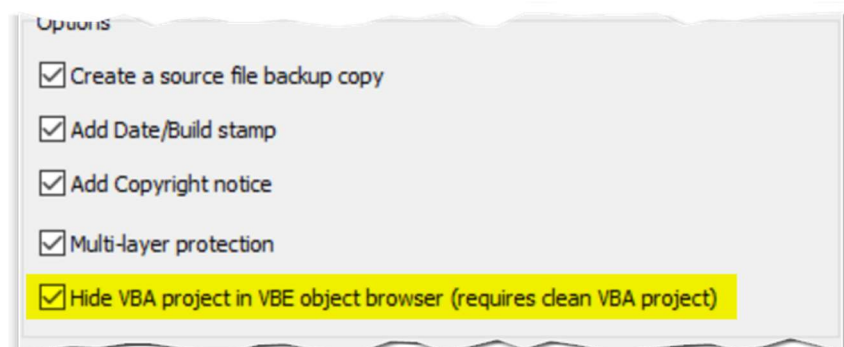
## [6] Clean & Compress VBA Code

File size can be reduced and code stability may be improved by using the VBA Code Cleaner powered by the Ribbon Commander framework. The Unviewable+ VBA application can detect the presence of the Ribbon Commander framework (RC), if available on your PC.



Uncleaned VBA projects can be cleaned using the RC VBA cleaner, before being locked. You will notice a file size reduction after VBA cleaning, which is normal.

**Use of a VBA Code Cleaner & Compressor is recommended for files destined for publication**, but is not required for VBA project protection, except for hiding projects in VBE Object Browser. Projects cannot be truly cleaned & compressed by merely exporting & importing all code modules.



Read more about the benefits of VBA code cleaning here:

<http://www.spreadsheet1.com/vba-project-code-cleaner-for-access-excel-powerpoint-word.html>

Case study: Benefits of code cleaning in the bloated Spinner v1.2 project

<http://www.spreadsheet1.com/flat-animated-preloaders-spinners-for-excel.html>

## Command line options (help switch: /?)

Files can be locked from the menu or using command line options.  
Wildcards are not supported.

```
Command line: Unviewable.exe file /pt:x /pl:x [/switches]
```

### Switches:

```
file ..... Filename for processing
/pt:x ..... Protection type
                x=0 ... Hidden Modules (Visible VBA Project)
                x=1 ... Unviewable password protected VBA project
                x=2 ... Unviewable locked VBA project
/pl:x ..... Protection level
                x=0 ... VbaDiff compatible (least secure)
                x=1 ... Simple
                x=2 ... Medium strength
                x=3 ... Ultimate (most secure)
/bak ..... Create a source file backup copy
/clean ..... Clean VBA project if dirty
/stamp..... Add Date/Build stamp
/cr ..... Add Copyright notice
/mlp ..... Multi-layer protection
/hp ..... Hide VBA project in VBE object browser
/enc ..... Encrypt file with default params
/pwd:password.... Password for locking/unlocking
```

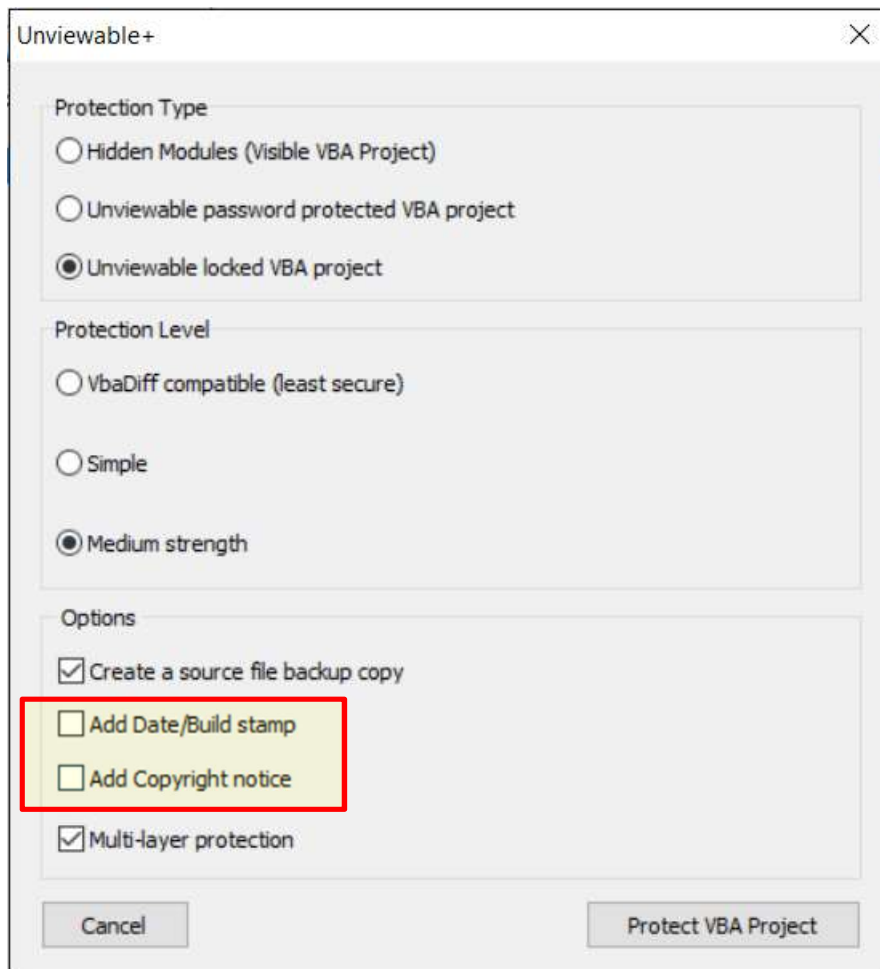


Please ensure that at least one **backup copy of your source code** is stored in at least one safe location, as the original commented VBA code cannot be recovered from obfuscated and locked Office projects.

## Word VBA protection

Unviewable+ can be used with Word up-to Medium level only (not with Ultimate), due some internal differences in Word VBA. Please **uncheck** the stamping & copyright options as shown below.

Code obfuscation and use of the DPB userform is strongly recommended in Word.



The image shows the 'Unviewable+' dialog box with the following settings:

- Protection Type:**
  - ☐ Hidden Modules (Visible VBA Project)
  - ☐ Unviewable password protected VBA project
  - ☒ Unviewable locked VBA project
- Protection Level:**
  - ☐ VbaDiff compatible (least secure)
  - ☐ Simple
  - ☒ Medium strength
- Options:**
  - ☒ Create a source file backup copy
  - ☐ Add Date/Build stamp
  - ☐ Add Copyright notice
  - ☒ Multi-layer protection

The 'Add Date/Build stamp' and 'Add Copyright notice' options are highlighted with a red rectangle, indicating they should be unchecked.

## Useful Tips from Unviewable+ VBA Academy

Unviewable+ VBA Academy is our latest initiative to help you improve the security features of your VBA applications and increase development productivity. These tips are emailed privately to our clients only. Please feel free to send us your feedback or your suggestions that may benefit the Unviewable+ developer community.

A commonly used attack is to disable application events (.EnableEvents=False) or hold the SHIFT key while opening a workbook/addin in order to prevent Workbook Open event code from running. Please review the workbooks that demonstrate the behaviour of the **OnLoad ribbon** event handler, among other potential useful tips:

- How to use a Windows timer in VBA to ensure Events are always enabled.
- How to hide data using legacy Macro worksheets that are not shown in VBE!
- Why OpenXML Strict spreadsheets offer stronger sheet/workbook protection.



To simplify and extend **ribbon UI** development from VBA, please learn about the Ribbon Commander framework for VBA & .NET

<http://www.ribboncommander.com/>

Additional articles:

<http://www.spreadsheet1.com/ribbon.html>

<http://www.spreadsheet1.com/vertex42-template-gallery-excel-addin-powered-by-ribbon-commander.html>

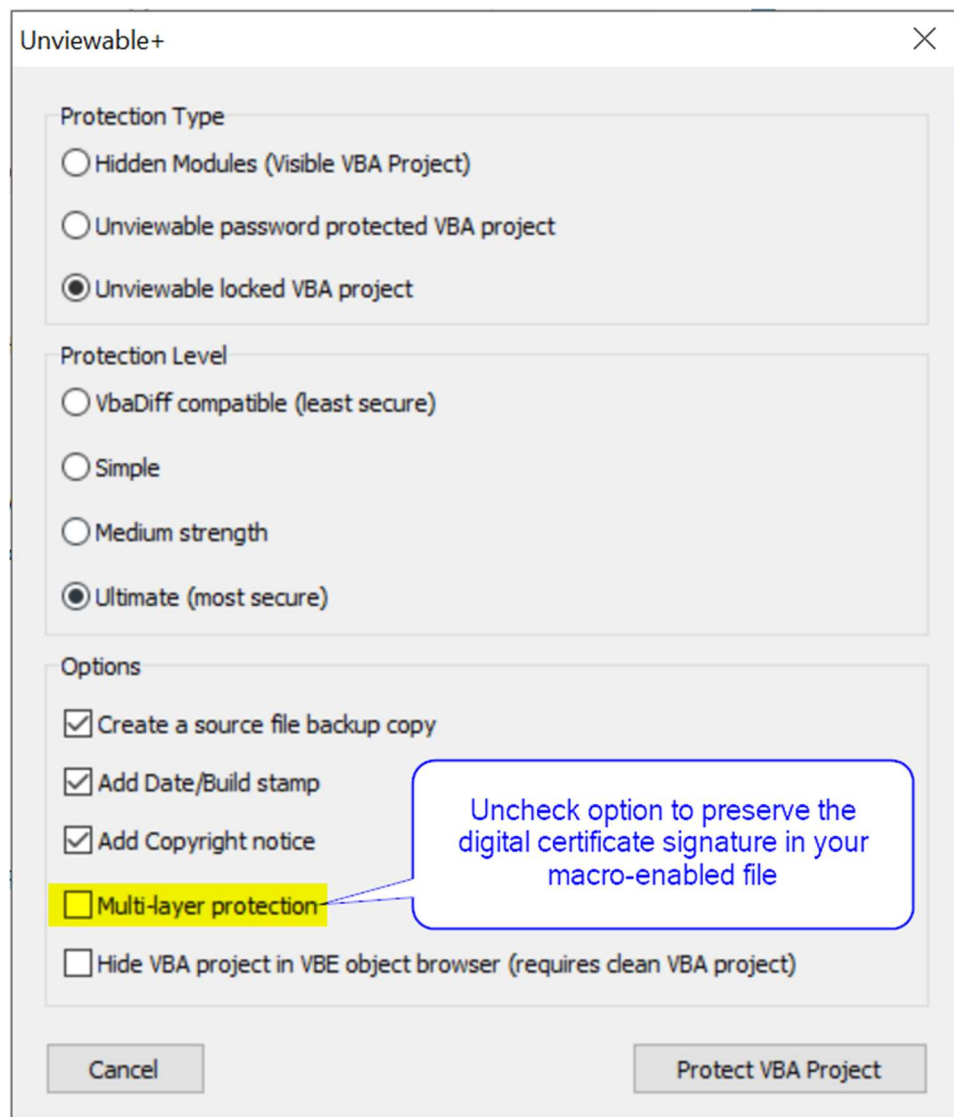
## Locking Files Signed With a Digital Certificate



You can sign files before or after locking VBA with Unviewable+ protection. However, the latter method is preferred since **Multi-layer protection** can be used.

If you opt to lock VBA in a project which is already signed with an digital certificate, please **uncheck the Multi-layer protection** option, otherwise the certificate will be discarded from the VBA project.

Unviewable+ will automatically uncheck the multi-layer option, if a signed VBA project is detected.



The screenshot shows the 'Unviewable+' dialog box with the following settings:

- Protection Type:**
  - ☐ Hidden Modules (Visible VBA Project)
  - ☐ Unviewable password protected VBA project
  - ☒ Unviewable locked VBA project
- Protection Level:**
  - ☐ VbaDiff compatible (least secure)
  - ☐ Simple
  - ☐ Medium strength
  - ☒ Ultimate (most secure)
- Options:**
  - ☒ Create a source file backup copy
  - ☒ Add Date/Build stamp
  - ☒ Add Copyright notice
  - ☐ Multi-layer protection
  - ☐ Hide VBA project in VBE object browser (requires clean VBA project)

A callout box points to the 'Multi-layer protection' checkbox with the text: "Uncheck option to preserve the digital certificate signature in your macro-enabled file".

Buttons at the bottom: Cancel, Protect VBA Project



A little known secret is that Office macros can be signed in an automated process, by using **Microsoft Office Subject Interface Packages** along with **SignTool**.

VBA projects can be signed manually from the command line or by using a batch file, without having to open any files in Office.

Unviewable+ locked VBA files can be signed with a digital code certificate without issues, even if the **multi-layer protection option** is checked. Please note that only macros in unencrypted Excel workbooks can be signed.

Please download and follow instructions in the bundled readme.txt file. Here are the Microsoft links:

SIPs: <https://www.microsoft.com/en-us/download/details.aspx?id=56617>

SignTool: <https://docs.microsoft.com/en-us/windows/desktop/seccrypto/signtool>

If you do not want to download the entire Windows SDK which is over 2GB in size, you can get the **SignTool** files from our cloud repository:

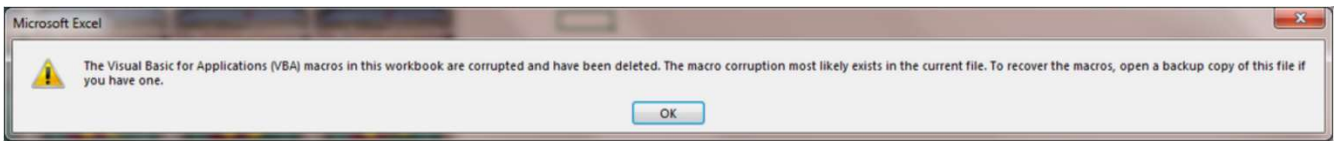
<https://spreadsheet1.box.com/shared/static/4fudxuuaiee8wvm3qsuhh5uctmojmw3y.zip>

### **Command line syntax:**

```
Signtool.exe sign /f cert.pfx /p myPassword /fd SHA256 D:\WBook.xlsm
```



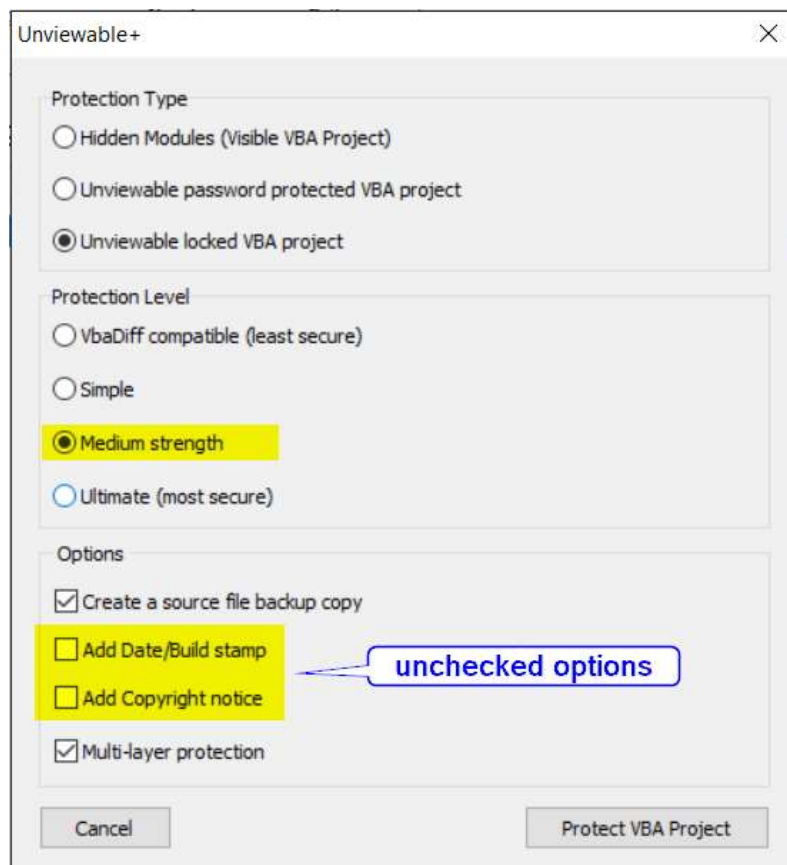
## Known Issues



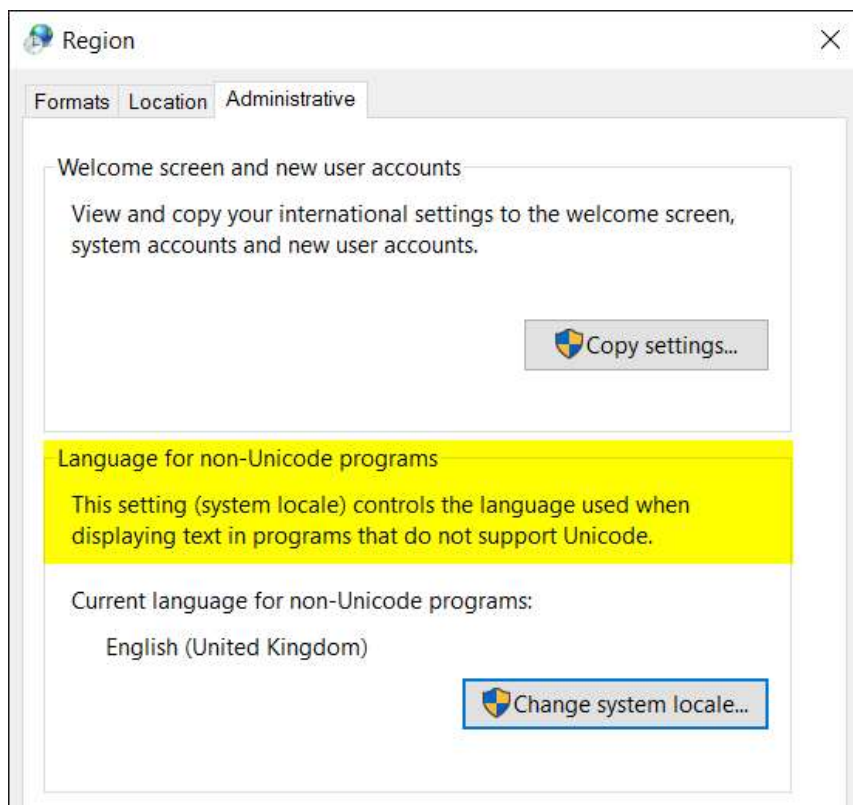
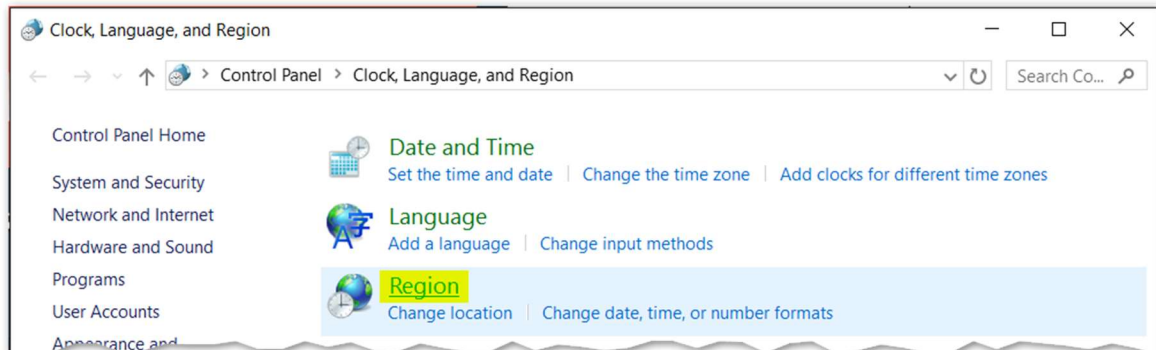
It has been reported that an unpatched version of Excel 2010 and an early build of Excel 2013 are not compatible with the Ultimate protection level and stamping options. Up-to-date Excel 2010 / 2013 versions can open Ultimate Unviewable+ locked files without any issues.

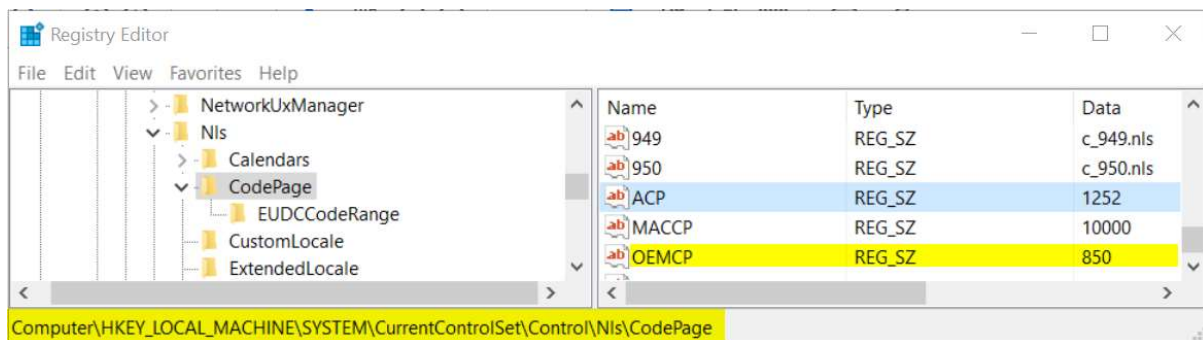
Another known issue is due to lack of Unicode support in VBE. Visual Basic Editor uses a legacy Windows technology called "code pages" to provide support for non-ASCII characters. The Ultimate protection level and stamping options are not compatible with some Code Pages, for example: Chinese, Czech, Polish, Russian, Slovak etc. However, Ultimate protection can be used with several non-English Code Pages.

- Please use 'Medium strength' protection and uncheck both date/build stamp and copyright notice, if the system locale for non-unicode programs is set to a language that triggers an error.



- An alternative workaround is to set the language for non-unicode programs to English (US or UK) ANSI code page: 1252, if Ultimate protection is required.



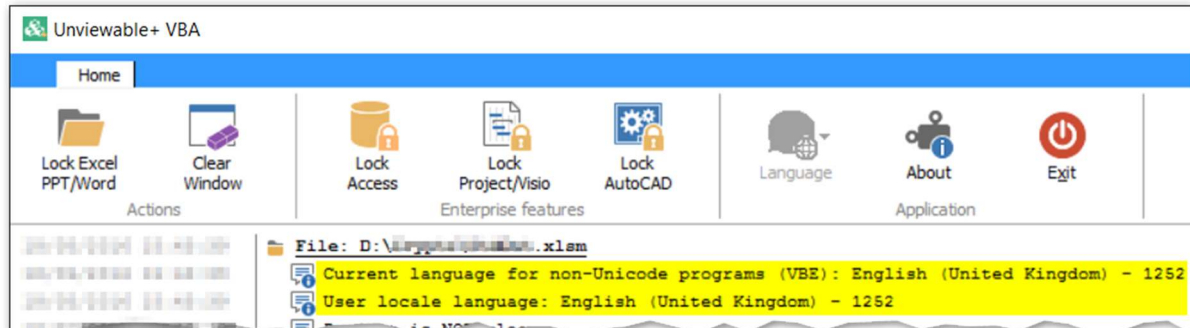


HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\CodePage\ACP

<http://www.spreadsheet1.com/how-to-display-foreign-characters-in-vbe.html>

The User Locale and the Code Page used in non-unicode programs (Visual Basic Editor) is reported by Unviewable+ VBA (version 2.78 or later) in the computer in which files are locked.

However, the Unviewable+ application does not modify the regional settings in PCs.



## When closing Excel, the <Project is Unviewable> dialog appears several times

This is a known but poorly-understood and scantily documented problem that affects a lot of innocent Excel files, but seems totally unrelated to Unviewable+. Several users have reported that the issue has been resolved after uninstalling the **Dropbox** Office app. Learn more from research published by Jon Peltier, Excel MVP:

<http://peltiertech.com/Utility20/Documentation20/VBPassword.html>

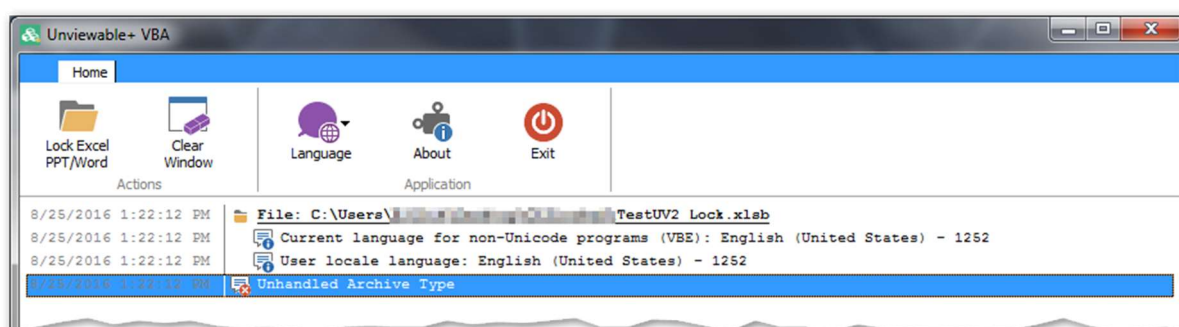
<http://peltiertech.com/vb-password-prompt-closing-excel/>

<https://www.dropbox.com/help/business/badge-on-off>

## How to work with Encrypted Office files

Office files saved in OpenXML format are ZIP containers that can be easily modified by the Unviewable+ application. However, files saved with a file-open password are encrypted and cannot be read!

You will get an "**Unhandled Archive Type**" or "**Encrypted file detected**" message, if you attempt to lock an encrypted Office file.



Please remove the password to open the file and the software will protect its VBA project. After running Unviewable+, you can add back the required password to open the file.

The same message appears, if your file has been significantly corrupted for some reason. Test, if the file can be opened without any corruption warning in Excel, PowerPoint or Word. Read more:

<http://www.spreadsheet1.com/how-to-diagnose-excel-file-corruption-and-repair-workbooks.html>

Office files can be encrypted/decrypted in bulk using a command line utility, without using Office. Learn more about encrypted Office files and Microsoft **DocRecrypt** tool:

<http://www.spreadsheet1.com/command-line-encryption-tool-for-office-excel-files.html>

## How to lock Office files destined for Mac: Use Medium level



Please use 'Medium strength' protection and uncheck both date/build stamp and copyright notice, if files locked are going to be used on Office for Mac.

Unviewable+

Protection Type

- ☐ Hidden Modules (Visible VBA Project)
- ☐ Unviewable password protected VBA project
- ☒ Unviewable locked VBA project

Protection Level

- ☐ VbaDiff compatible (least secure)
- ☐ Simple
- ☒ Medium strength
- ☐ Ultimate (most secure)

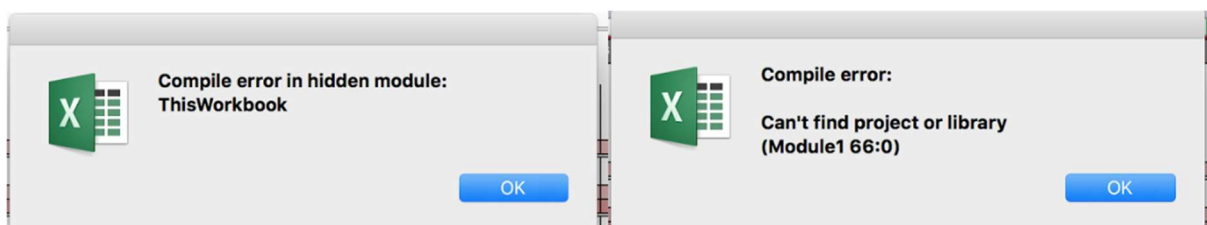
Options

- ☒ Create a source file backup copy
- ☐ Add Date/Build stamp
- ☐ Add Copyright notice
- ☒ Multi-layer protection

Cancel Protect VBA Project

unchecked options

**Warning:** Your VBA code may break with errors as shown below, due to Mac/Windows compatibility issues and failure to load certain libraries only found in PCs. Please ensure that any libraries used in your code with either early (referenced) or with late binding exist on Mac computers.

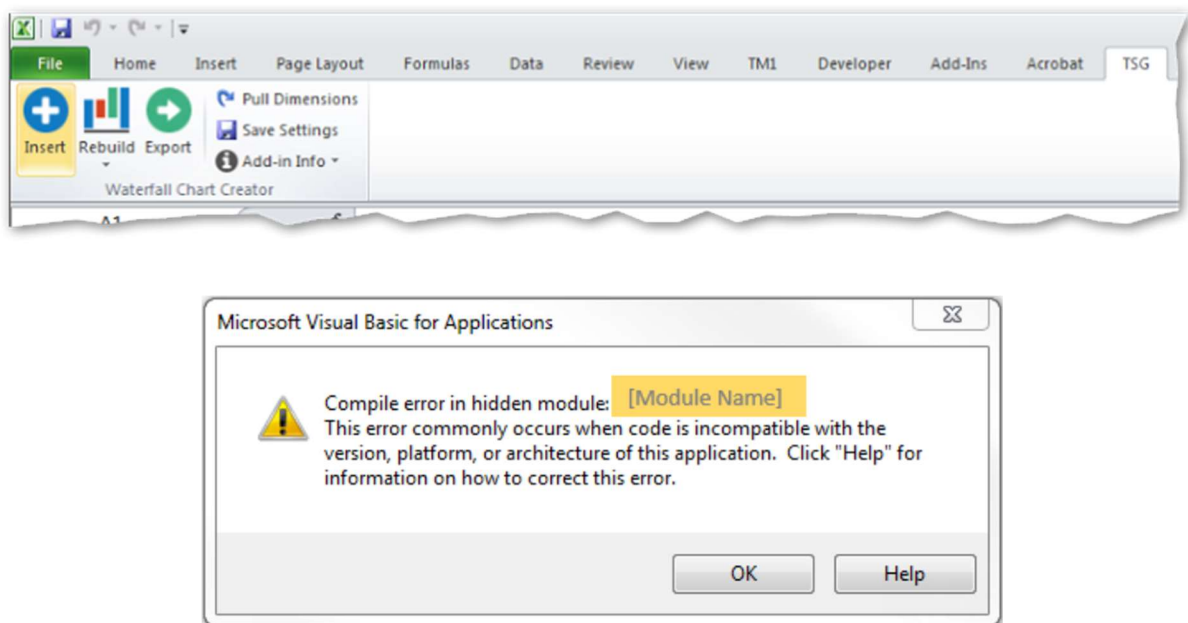


## VBA Warning: Compile error in hidden module

Your VBA code may not be backwards compatible with some Excel versions or a VBA reference is missing from the target machine, if:

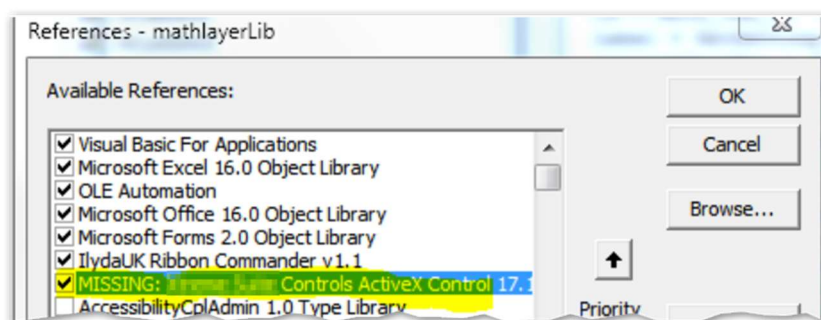
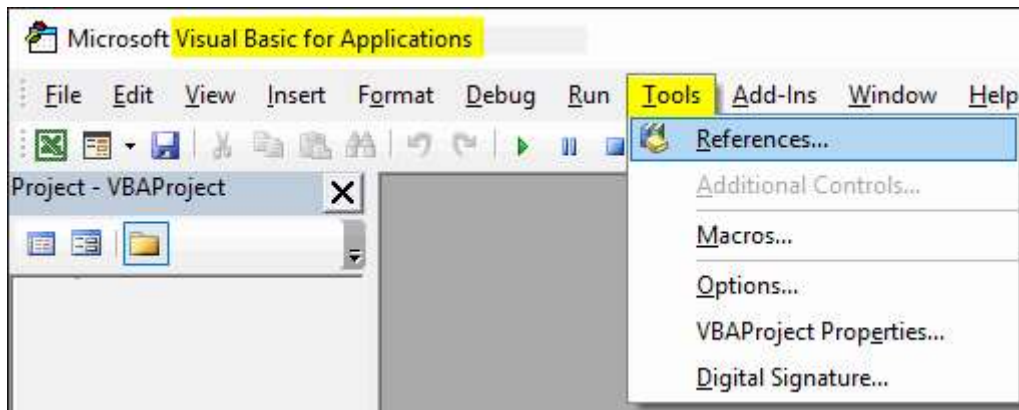
[1] A message as shown below appears during workbook or addin use e.g. when a ribbon button is clicked.

[2] The workbook or addin loads normally e.g. ribbon UI is shown

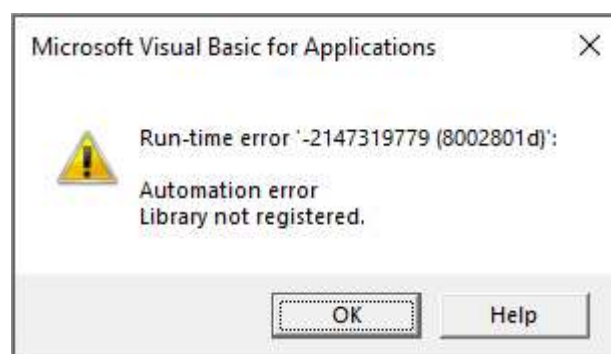


If a user is receiving the above message, it means that there is VBA code in the called out Module that is incompatible with the user's version of Excel. For example, if you use the method **FullSeriesCollection** in your code and the user is using Excel 2010 or prior, they will receive this message. This is because FullSeriesCollection was first created with the release of Excel 2013 and is not backwards compatible.

To check your code, please compile your unlocked VBA source code (use the Debug menu in VBE) in every Excel and Windows version that you plan to use and ensure that any 3<sup>rd</sup> party libraries used are available at user machines.



Please verify that all libraries referenced in VBA projects you plan to lock with Unviewable+ exist on the computers of your users or clients, otherwise an error will be shown at runtime and code execution will stop.



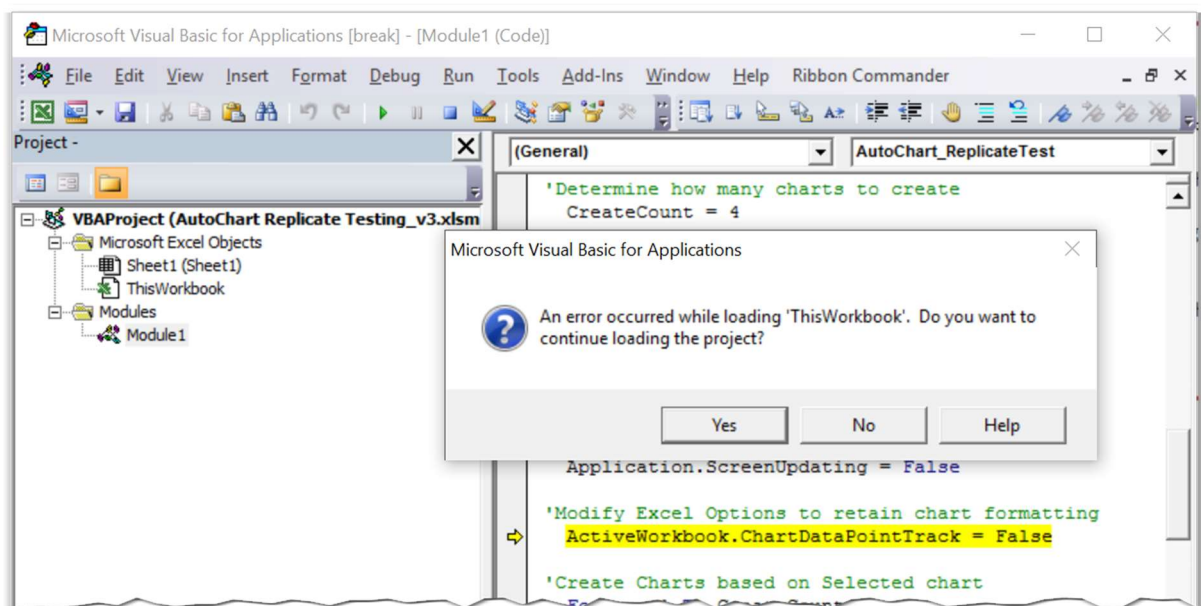


## No intellisense for UDFs in Hidden modules or Ultimate protection

It is a known issue that UDF intellisense will be no longer available in macro-enabled files that have VBA function code stored in **hidden modules** (visible project) or in files that have been locked with the **Ultimate** level.

## Workbook loading error while VBE is in Debug mode

Please click the RESET button in VBE, to exit debug mode, before opening a macro-enabled file ( VBE > Run > Reset )



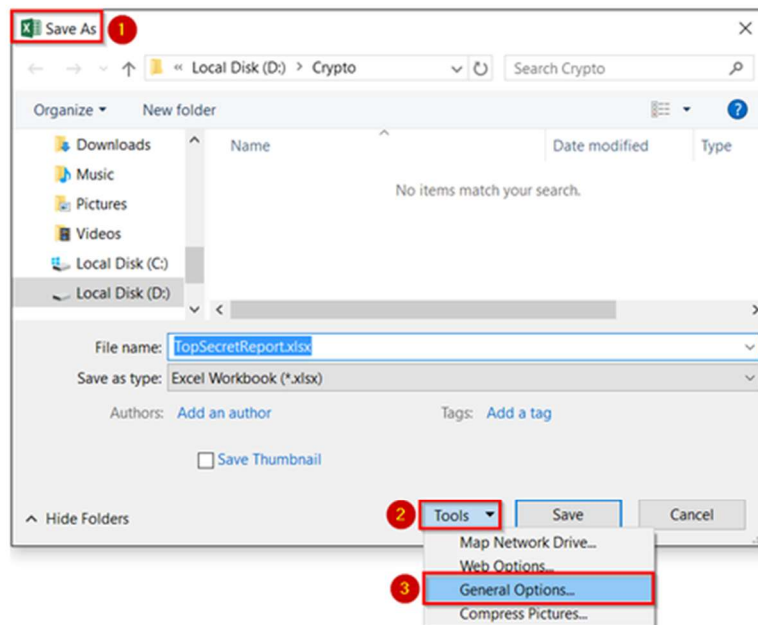


## Use AES Encryption to protect against attacks on closed files

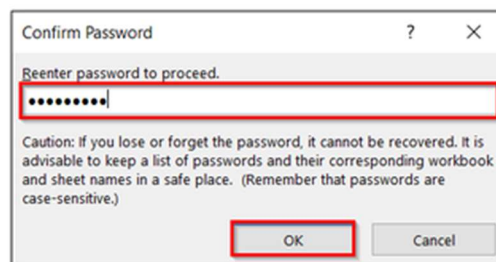
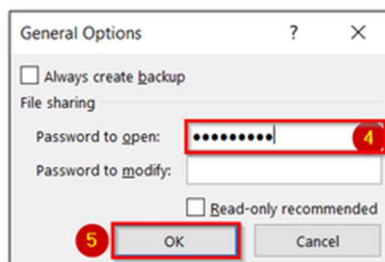
Excel files can be password-encrypted, meaning that it can be impossible for most hacking programs (e.g. password crackers, hex editors) to read the file contents, including its VBA code. Here is a quick reminder how to set a file-open password:

### How To Password Protect Opening Workbooks in Excel

Open a workbook, navigate to the **Save As** dialog box (File > Save As > Browse), then click the **Tools** drop-down menu near the bottom of the dialog box. Open the Tools menu & select **General Options**



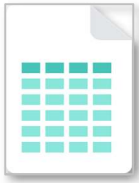
Type in your desired **password** in the password-to-open field. Click **OK**. You will be asked to verify your password by entering it in again. Now, your file has an encryption password to prevent unauthorized people from opening up and viewing your Excel workbook.



Finally, click the **Save** button to convert your file in an encrypted format using the AES algorithm.

Obviously, the problem is that users must enter a password to open encrypted files! However, there is an undocumented Office method to encrypt a workbook without Excel asking for a password when opening the file.

The reason that Excel does not ask for a password is because the file has been encrypted with the **default Excel password** which is **VelvetSweatshop**. Whenever Excel opens an encrypted file, it automatically tries the password VelvetSweatshop first. The above trick applies to Excel only (neither for PowerPoint, nor for Word).



**Workbooks:** The password VelvetSweatshop is not retained when saving a workbook. Please use a Workbook\_BeforeSave event to force saving the file with the default Office encryption password, otherwise your workbook will be saved in an unencrypted format.

```
Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)
```

```
    ' The default Office encryption (file-open) password does NOT persist during  
    Save/SaveAs in workbooks
```

```
    ' The code below is NOT needed for add-ins
```

```
    ' The workbook must be encrypted AFTER VBA is locked in an unviewable state
```

```
    ' Prevent BeforeSave event from firing when the workbook is saved in your PC
```

```
    If GetDriveSerial("C") = "D4666A11" Then
```

```
        ' Don't encrypt workbook while VBA is unlocked - my disc serial detected
```

```
        ThisWorkbook.Password = ""
```

```
    Else
```

```
        ' Default Excel file-open password
```

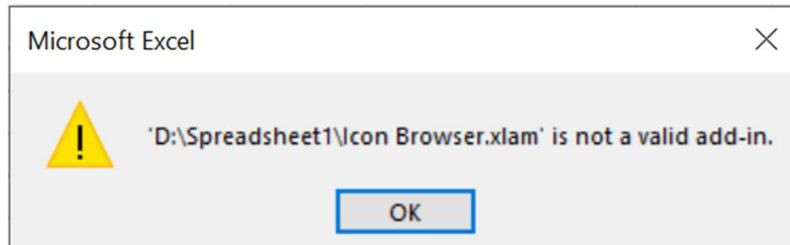
```
        ThisWorkbook.Password = "VelvetSweatshop"
```

```
    End If
```

```
End Sub
```



**Add-ins:** Excel add-ins cannot be saved in an encrypted state by Excel from the SaveAs menu. In addition encrypted add-in files cannot be installed as native add-ins in Excel. Workaround:



To encrypt add-ins, please use the free, open source command line tool developed by security expert and Microsoft MVP Shigeo Mitsunari from Cybozu Labs, Inc, Japan. Read more:

<http://www.spreadsheet1.com/command-line-encryption-tool-for-office-excel-files.html>

Copy the command line binary from **bin** Github folder: msoffice-crypt.exe (no installation required). Github project page: <https://github.com/herumi/msoffice>

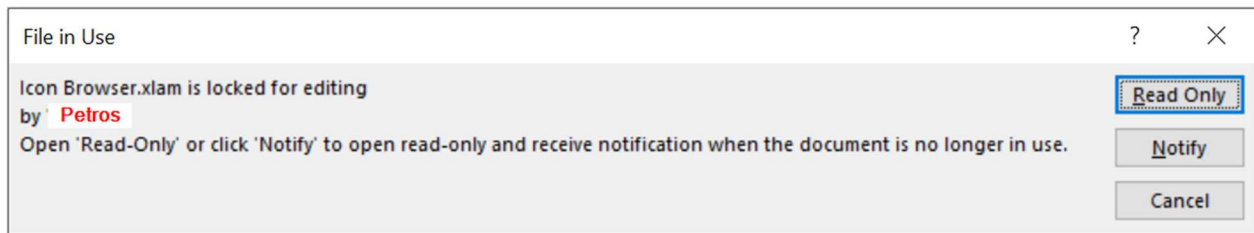
Alternatively, you can download the executable signed with our digital certificate.

Syntax: **msoffice-crypt.exe -e -p VelvetSweatshop <path - filename>**

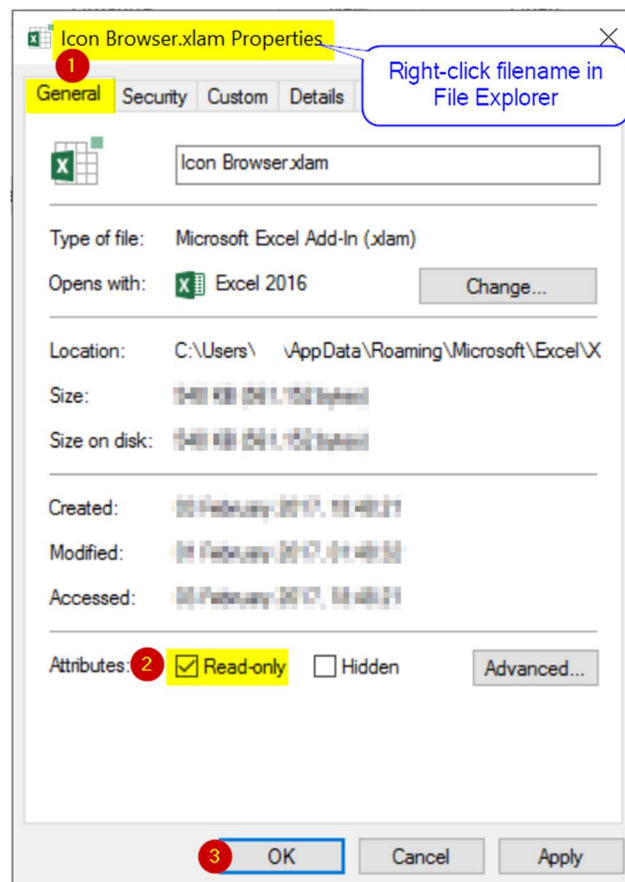
Encrypted add-ins cannot be opened by Excel, if installed in the default Excel folders. To overcome this issue:

- Open encrypted add-ins from any folder other than the default, version agnostic add-in folder: **%appdata%\Microsoft\AddIns**
- Open encrypted add-ins from the XLSTART version agnostic add-in folder, in which users have read-write access: **%appdata%\Microsoft\Excel\XLSTART**

However, when an add-in saved in XLSTART folder is opened in one Excel version, opening it in another results in an Open Read Only message.



To avoid this message when several Office versions have been installed side-by-side, set the Read-only file attribute or save the add-in as Shared file.



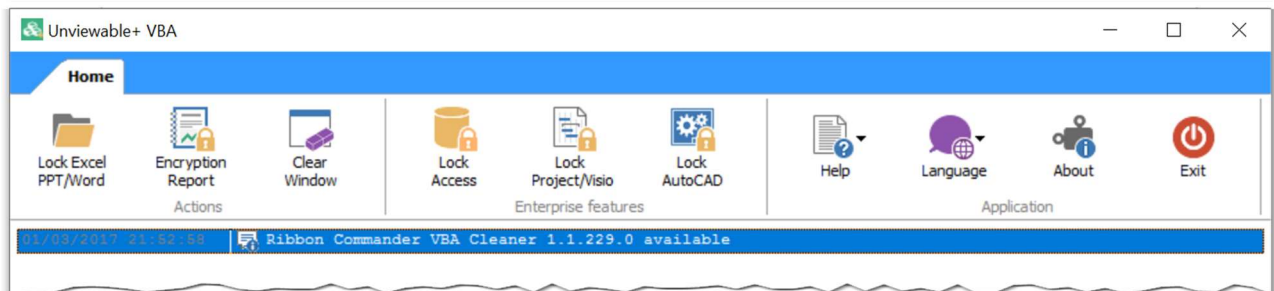
- Use a helper add-in and open the encrypted add-in using the **Workbooks.Open** method from a normal VBA add-in. Please omit the password argument to avoid tipping off hackers that the file is a native encrypted Excel add-in.

To confuse hackers even more:

[1] Rename your VBA project as DLL (screenshot shows an unlocked [viewable] file for demonstration purposes)

# Unviewable+ for Access, Project, Visio & AutoCAD

## ENTERPRISE FEATURES FOR SITE & GLOBAL LICENSES



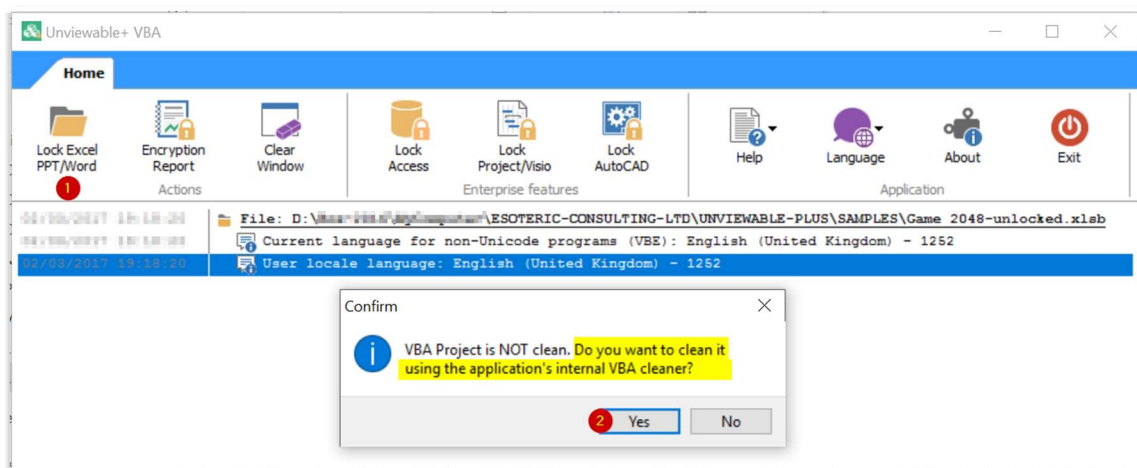
Site and Global licenses offer protection support for:

- Microsoft Access (simple locked level only)
- Microsoft Project (without multi-layer protection)
- Microsoft Visio
- Autodesk AutoCAD (without multi-layer protection, up-to Medium level)

The application licensed to 1 Developer cannot be run on the **Windows Server** operating system. Please purchase a **Site** or **Global** license, if you need to run the Unviewable+ application on Windows Server.



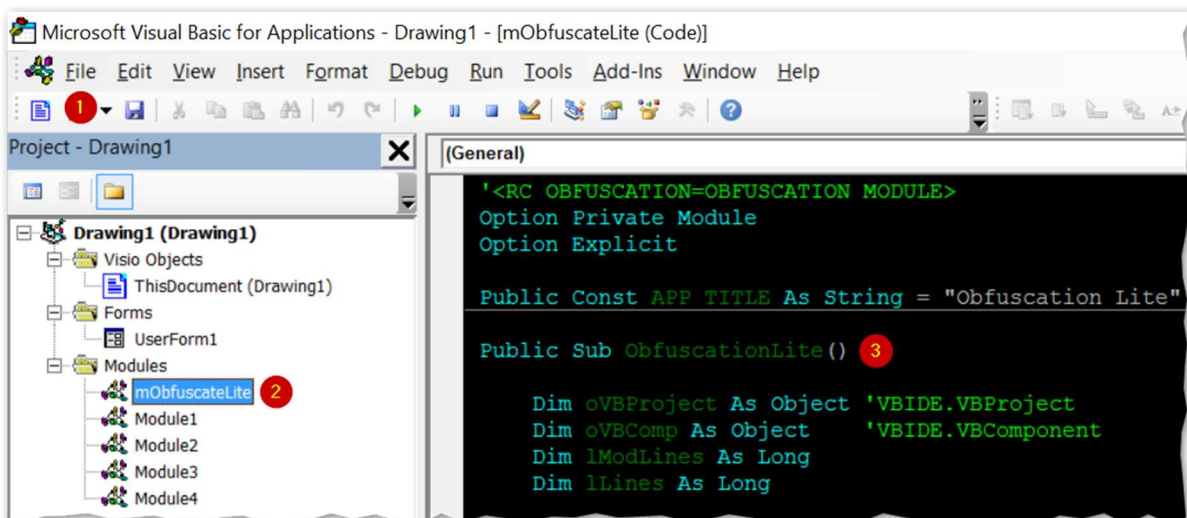
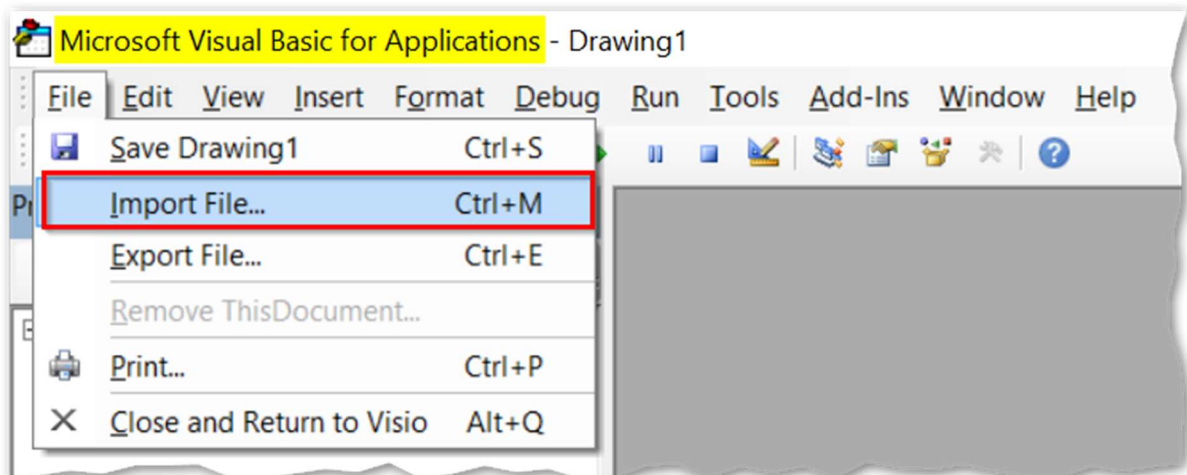
The Site & Global licenses offer an **internal VBA code cleaner**, which allows users to clean their macros to increase stability and hide VBA projects in VBE Object Browser.



## Source Code for Code Obfuscation

Customers who purchase Site or Global licenses will receive the obfuscation module (lite) in VBA source code (.bas), since add-ins are not available for Access, Project, Visio & AutoCAD. Reference to the VBE Extensibility library is not required, as the code is written using late binding.

1. Import the mObfuscation.bas module into the active unlocked VBA project.
2. Select the **mObfuscationLite** module
3. Run the ObfuscationLite procedure. The redundant obfuscation module will be automatically deleted, after your VBA project has been obfuscated.
4. Compile your VBA project again to verify that obfuscation did not break your program, before saving it.



## Global License: VBA Tools Add-in Source Code

Corporate customers who purchase a Global license will receive the source code of the entire VBA Tools add-in.

The Delphi source code is not included in the global license by default, but is available for purchase for an extra fee by corporate clients with a global license. Please [contact us](#) for more info.



Source code is intended for enterprise use only and not for resale.



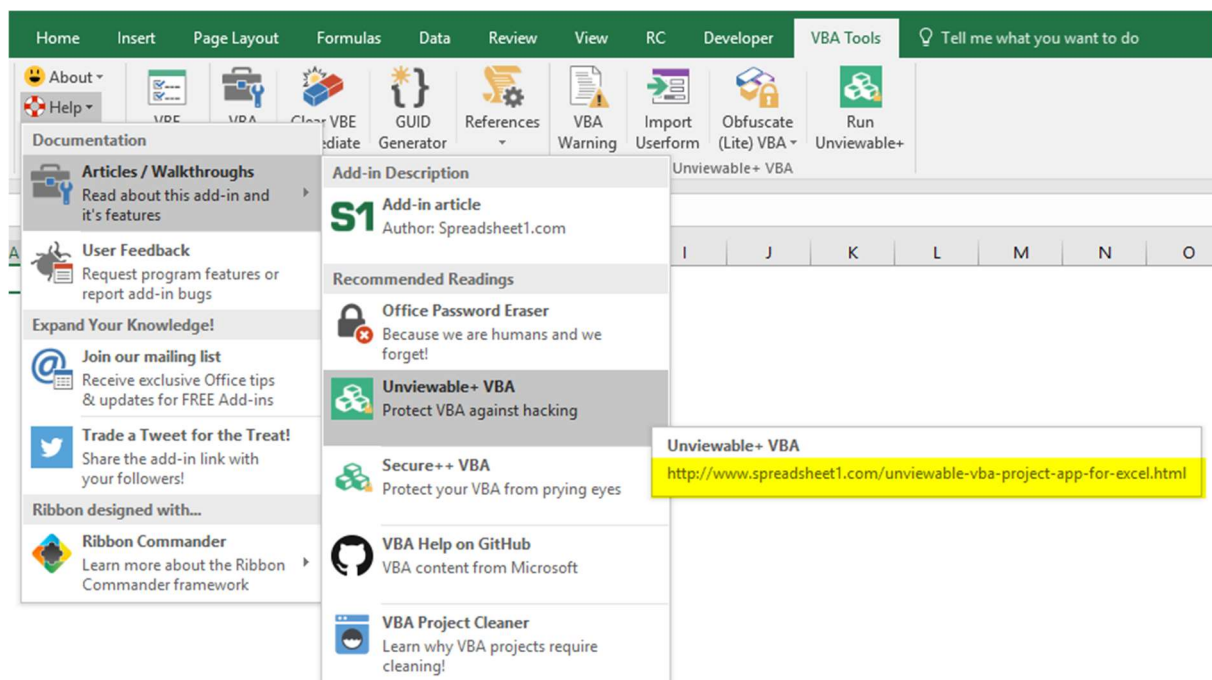
## Notes on Customer Privacy

The Unviewable+ application and companion VBA add-in do not require a connection to the internet to ease any concerns from customers who are sensitive about their privacy or/and about corporate espionage.



The Unviewable+ Delphi EXE is a portable application, does not require installation, does not have a licensing module, so internet is never needed. The only licensing info is in the customized About form, which shows the name of customer (logo is optional).

Along the same lines, the companion **VBA Tools** addin does not connect to the internet behind the scenes, neither any licensing, nor any complex installation is required. Some ribbon buttons launch web articles published in our website, but usage is optional and the links are shown in ribbon screentips.



- The protection scheme is 100% passive, no code runs when locked Office files are opened.
- We do NOT offer a VBA code recovery service for unviewable or custom protected Office files.

## Frequently Asked Questions

**Is it best to protect the VBA project with a password before making it Unviewable, or should there be no password protection on VBA project when running the Unviewable application?**

A normal VBE password neither increases, nor decreases the protection strength of macro-enabled files locked with the Unviewable+ application. Please feel free to use a normal VBE password, if that suits your work process.

**If I go to VBIDE and click on the project, the message “Project is Unviewable” appears and I click OK and then it appears AGAIN. Is that normal?**

No, the Project is Unviewable should appear only once.

**I made an Excel workbook unviewable and it seems fine until I go to close the file. At that point I get the message “Project is Unviewable” , then Excel crashes and tries to reload.**

It has been reported that this issue is resolved after disabling the **Dropbox Badge**.

**If I have installed Unviewable+ on my computer and I clone my hard drive, will the installation of the program on my cloned copy work, or will I have to get it activated again?**

Yes, Unviewable+ will run from the cloned drive without activation, as there is no licensing module in our app in order to ensure customer privacy.

**I am trying to find a tool that will enable me to continue even if the author goes out of business. Can I use Unviewable+ if your company ceases to exist?**

Yes, there is no licensing module in our app, so software activation is never needed!

## Appendix: Permanent Links

For the latest public documentation updates, please visit the Unviewable+ web page:

<http://www.spreadsheet1.com/unviewable-vba-project-app-for-excel.html>

### Test files:

<https://spreadsheet1.box.com/shared/static/evdm2py0mgry61kue9500jw5uupoidrd.zip>

### VBA Tools Add-in:

<https://spreadsheet1.box.com/shared/static/lyjsw2lcuwl5uvur7ugczsu3hvfs2awl.xlam>

### Encryption (msoffice-crypt project):

<https://spreadsheet1.box.com/shared/static/0y1shdaxcpw8rxm8ah5mgf64hvb8tpa.zip>

**PDF:** <https://spreadsheet1.box.com/shared/static/qu01kjb7v0nrthaqwdq58jjrqnsatl.pdf>

At **Spreadsheet1.com** our goal is to continually improve our VBA protection solutions. We are committed to making sure our software is effective, easy to use and closely aligned with the dynamic and evolving needs of VBA developers. But we can't do it without your help.

To help us improve our software or to report a problem please do not hesitate to contact me: [petros@spreadsheet1.com](mailto:petros@spreadsheet1.com)

