Apex Support Bulletin: Office 2016 and 2019 Visual Basic Security Controls

Revision 1.2 [December 5, 2018]

Contact eriksc@microsoft.com

Summary

Office 2016 and 2019 for Mac has a number of preference settings that administrators can manage through MDM configuration profiles to control and enforce the level of access to the local Mac for Visual Basic macros in Office documents.

Visual Basic Security Controls

All of the below settings are read from and managed in the 'com.microsoft.office' preferences domain, which the OS stores in the ~/Library/Preferences/com.microsoft.office.plist file. These preferences are suite-wide; each setting applies to all of the applications in the suite (note that only Word, Excel, and PowerPoint support Visual Basic for Applications). There is no way to apply a different preference value to an individual application.

These settings exist in both Office 2016 and 2019:

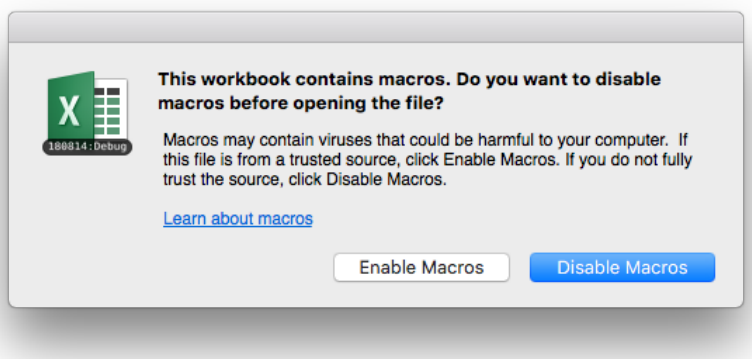
Setting Name	As of Version	Type	Possible Values	Default Value
VisualBasicMacroExecutionState	15.32	String	DisabledWithoutWarnings DisabledWithWarnings EnabledWithoutWarnings	DisabledWithWarnings
DisableVisualBasicExternalDylibs	15.31	Boolean	YES / NO	NO
AllowVisualBasicToBindToSystem	15.31	Boolean	YES / NO	NO
DisableVisualBasicToBindToPopen	16.16	Boolean	YES / NO	NO
DisableVisualBasicMacScript	16.16	Boolean	YES / NO	NO

These settings exist in Office 2019 only:

Setting Name	As of Version	Type	Possible Values	Default Value
VBAObjectModelIsTrusted	16.21	Boolean	YES / NO	NO

VisualBasicMacroExecutionState

The VisualBasicMacroExecutionState[1] setting controls whether macros are ever allowed to execute, and what the user experience is when opening a file that contains a macro. By default, macros will only run after the user is presented with an alert when opening a file that contains a macro. The user must make a choice whether to allow or deny macros for each individual file, each time that file is opened.



Users have the ability to change this setting for themselves in the Preferences UI for each Office for Mac application that supports Visual Basic, but any admin setting deployed via an MDM configuration profile will override the user setting and will disable the setting in the UI.

Administrators can suppress the alert and opportunity for user choice by specifying one of the other settings:

- DisabledWithoutWarnings — (most secure) The alert is suppressed, and macros are not allowed to run
- DisabledWithWarnings — (default) The alert is shown, and the user makes the choice to enable or disable macros in that one file
- EnabledWithoutWarnings — (least secure) The alert is suppressed, and macros are allowed to run

DisableVisualBasicExternalDylibs

The DisableVisualBasicExternalDylibs setting determines if macros are allowed to use a DECLARE[2] statement to bind a Visual Basic symbol name to an external procedure in the local OS. The default value for this setting is to allow binding to external dylibs, because many legitimate 3rd party addin vendors use this feature of Visual Basic to add and extend features in Office for Mac. When this setting is set to NO, macros that attempt to use a DECLARE statement will fail with an error at the point where the external procedure is invoked.

AllowVisualBasicToBindToSystem

The AllowVisualBasicToBindToSystem setting determines if macros are allowed to use a DECLARE to bind to the system() OS API. This API allows macros to execute arbitrary external processes and pass them arbitrary data on the command line. The default value for this setting disallows the binding, as the system() API should not be used[3]. When this setting is set to NO, macros that attempt to use system() will fail with an error at the point where system() is invoked.

---

[1] https://docs.microsoft.com/en-us/deployoffice/mac/set-preference-macro-security-office-for-mac
[2] https://docs.microsoft.com/en-us/office/vba/Language/Reference/User-Interface-Help/declare-statement
[3] https://wiki.sei.cmu.edu/confluence/pages/viewpage.action?pageId=87152177

AllowVisualBasicToBindToPopen

The DisableVisualBasicToBindToPopen setting determines if macros are allowed to use a DECLARE to bind to the popen() OS API. This API allows macros to execute arbitrary external processes and pass them arbitrary data on the command line. The default value for this setting allows the binding, as at least one 3rd party vendor uses popen to communicate with their own code. When this setting is set to NO, macros that attempt to use popen() will fail with an error at the point where popen() is invoked.

DisableVisualBasicMacScript

The DisableVisualBasicMacScript setting determines if macros are allowed to invoke the MacScript[4]() Visual Basic API. This API allows macros to execute arbitrary processes via AppleScript by including "do shell script …" in the AppleScript code. The default value for this setting allows using MacScript, as there are a number of legitimate uses of AppleScript that do not rely on external processes. When this setting is set to NO, macros that attempt to use MacScript will fail with an error at the point where MacScript is invoked.

VBAObjectModelIsTrusted (Office 2019 only)

The VBAObjectModelIsTrusted setting determines if macros are allowed to modify the VB project itself through the VBA object model. The default value for this setting distrusts the VB object model and prevents macros from modifying the VB project. When this setting is set to NO, macros that attempt to invoke any method in the VB object model will fail with an error at that point in the code.

Discussion

These settings allow administrators to customize the level of access that VB macros have on the local Mac. Administrators should be aware of user reliance on 3rd party addins that make use of Visual Basic to extend Office for Mac, as disabling some Visual Basic functionality with these settings may adversely affect the functionality of these 3rd party products.

Document History

Date/Version	Changes
August 21, 2018 – 1.0	Initial version
August 27, 2018 – 1.1	Added notes on where prefs are stored and footnote for VisualBasicMacroExecutionState
December 5, 2018 – 1.2	Added VBAObjectModelIsTrusted information

[4] https://docs.microsoft.com/en-us/office/vba/Language/Reference/User-Interface-Help/macscript-function