

# **MZ-Tools 8.0**

**Top 10 advanced features of MZ-Tools**

**<https://www.mztools.com>**

**Carlos Quintero**

**AEUG. February 2024**

# About me

- Developer in VB “Classic”, VB.NET and C# since 1995
- Former Microsoft MVP (Most Valuable Professional) 14 times (2004-2018)
- Developer of MZ-Tools for VBA, VB6 and Visual Studio
- Currently DevOps architect (CI/CD pipelines, etc.)



## Contact:

- [mztools@mztools.com](mailto:mztools@mztools.com)
- <https://www.linkedin.com/in/carlosquinteroivivar/>

# About MZ-Tools

- Add-in / plug-in to provide productivity tools to IDEs
- <https://www.mztools.com>
- Perpetual license (no annual fee)
- Unlimited support included (no support fee)
- All your IDEs: VBA, Visual Studio, VB6
- Licensed per person, all your computers (home, work, VM, cloud...)
- Microsoft Office 64-bit & 32-bit
- 30-day trial, 100% money back guarantee
- Regular price Single license: \$79.95
- Pack3, Pack5, Pack10, Site, etc.
- 50% discount coupon (until Apr 30 2024): AEUG2024

**MZ-Tools**

Home MZ-Tools 8.0 Features Download Buy Documentation Changelog Support About

## MZ-Tools 8.0

for VBA (64-bit & 32-bit), Visual Studio (.NET) and VB6

One perpetual license. All your IDEs. All your computers.

**Supported IDEs**

- VBA Editor of Microsoft Office (64-bit & 32-bit)  
(Office 365, 2021, 2019, 2016, 2013, 2010, 2007, 2003, 2002, 2000)
- VBA Editor of 3rd party applications  
(Autodesk Inventor, AutoCAD, SolidWorks, CATIA, CorelDRAW, etc.)
- Microsoft Visual Studio 2022
- Microsoft Visual Studio 2019
- Microsoft Visual Studio 2017
- Microsoft Visual Studio 2015
- Microsoft Visual Studio 2013
- Microsoft Visual Studio 2012
- Microsoft Visual Studio 2010
- Microsoft Visual Basic 6.0

**Supported Languages**

- Visual Basic for Applications (VBA)
- Visual C#
- Visual Basic .NET
- Visual Basic 6.0

Features | Documentation | Buy

Download

# About the MZ-Tools name

- In the year 2000 the original domain qtools.com (Quintero's Tools) was already taken
- Mazinger-Z: very powerful robot of a very popular Japanese cartoon at Spain in late 1970's
- Domain mztools.com (Mazinger-Z's Tools) was available...

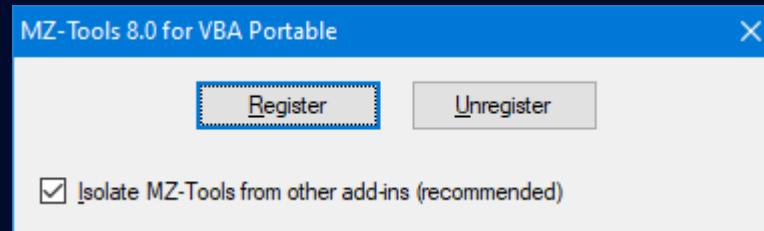


# Top 10 advanced features

- 1. Portable edition
- 2. Customizing MZ-Tools buttons on built-in toolbars and new toolbars
- 3. Keyword expansion for code templates
- 4. Development Environment Settings Review
- 5. Generate custom HTML documentation from XML documentation with XSLT
- 6. Find / Replace with regular expressions
- 7. Custom code element sorting
- 8. Nomenclature Rules Review
- 9. Programming Rules Review
- 10. Automating MZ-Tools features with VBScript / PowerShell

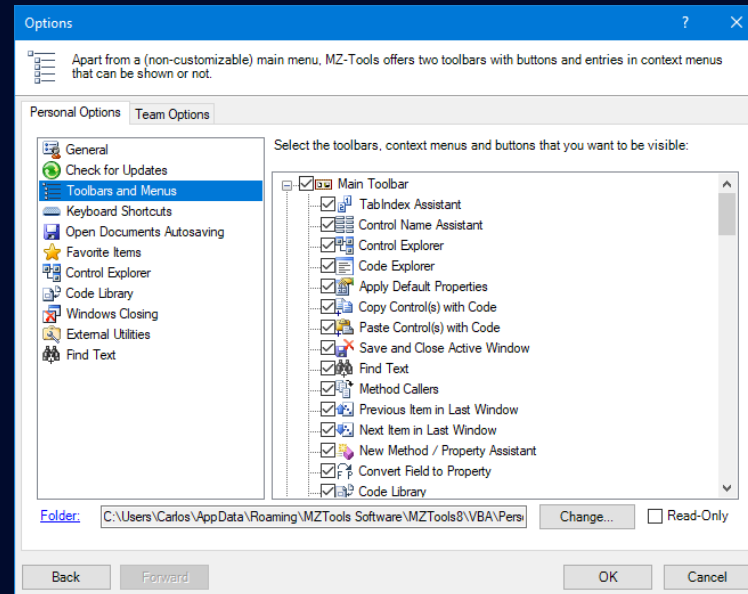
# 1. Portable Edition

- The setup-based edition doesn't not require admin rights
- The portable edition is a last resort when you are not even allowed to run setups that don't require admin rights
- Runnable from external drive
- Small utility to create the minimal entries in the Windows Registry

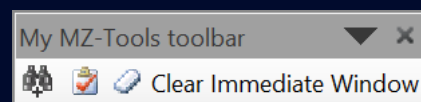


## 2. Customizing MZ-Tools buttons

- The Toolbars and Menu option page allows to make toolbars / buttons visible or invisible:

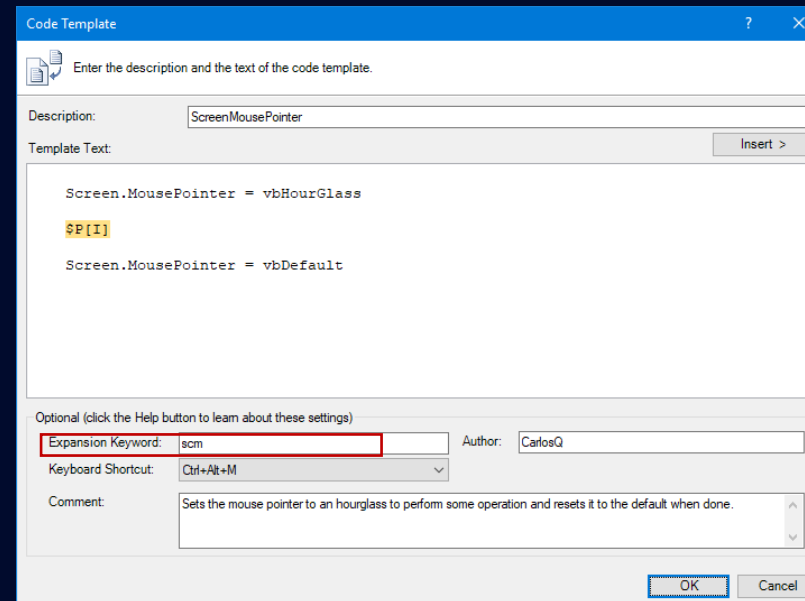
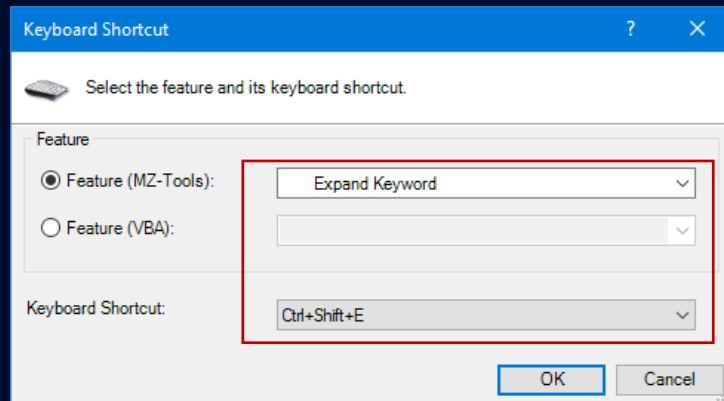


- But you can also put buttons on built-in toolbars or create custom toolbars:



# 3. Keyword expansion for code templates

- Keyword that, with a keyboard shortcut (associated to the Expand Keyword command) expands into a code template.



```
Private Sub Form_Load()  
scm|  
End Sub
```



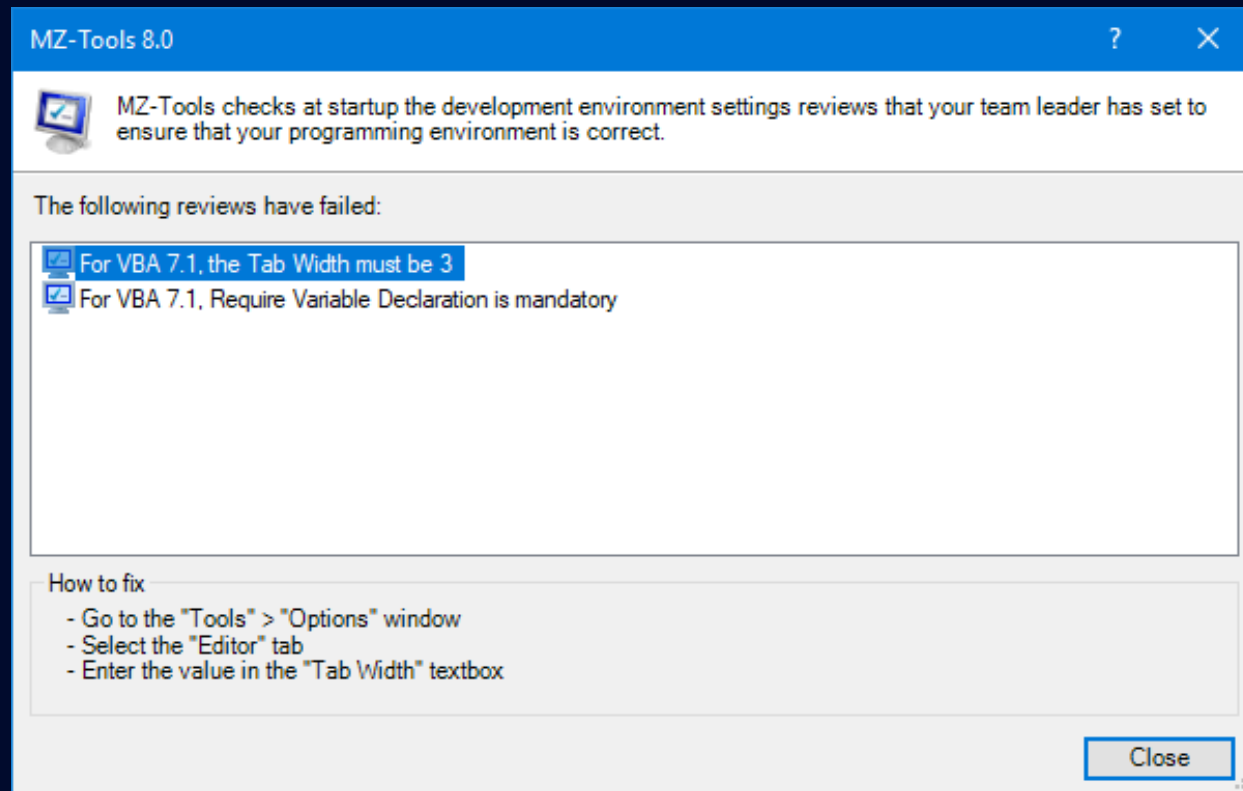
Ctrl + Shift + E

```
Private Sub Form_Load()  
  
Screen.MousePointer = vbHourglass  
|  
  
Screen.MousePointer = vbDefault  
  
End Sub
```



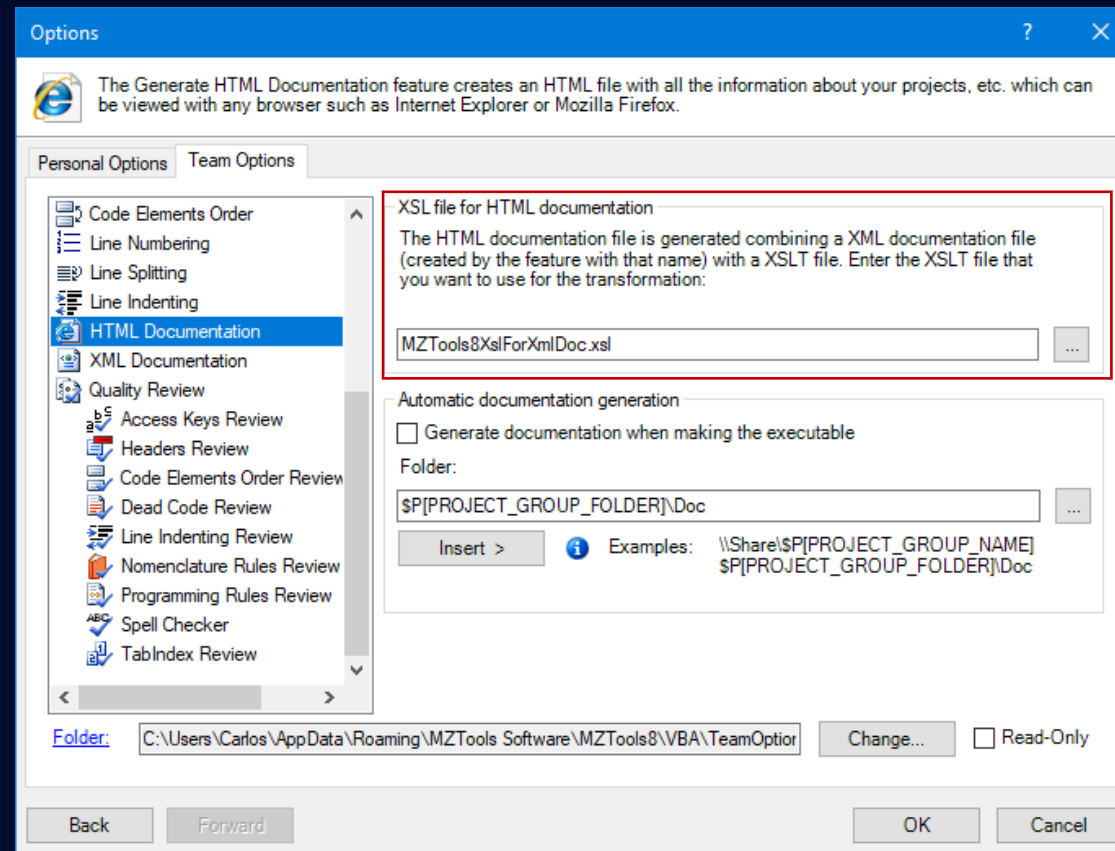
# 4. Development Environment Settings Review

- Enforce environment configuration on a team (or multiple computers of a single person)



# 5. Generate custom HTML documentation

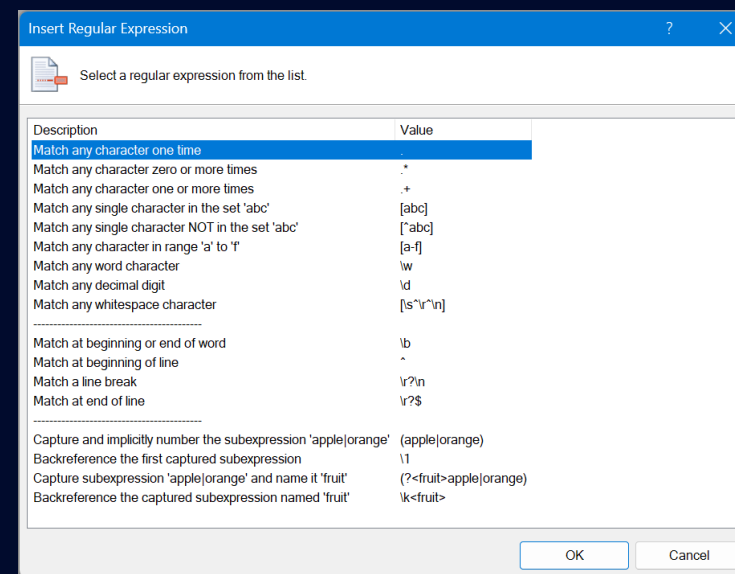
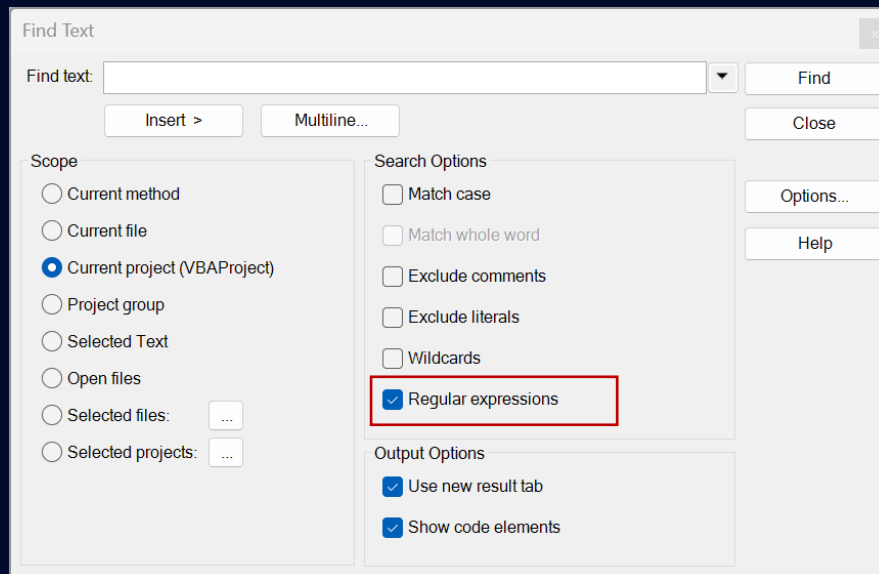
- Customize the XSLT file used to transform XML Documentation into HTML Documentation



# 6. Find / Replace with regular expressions

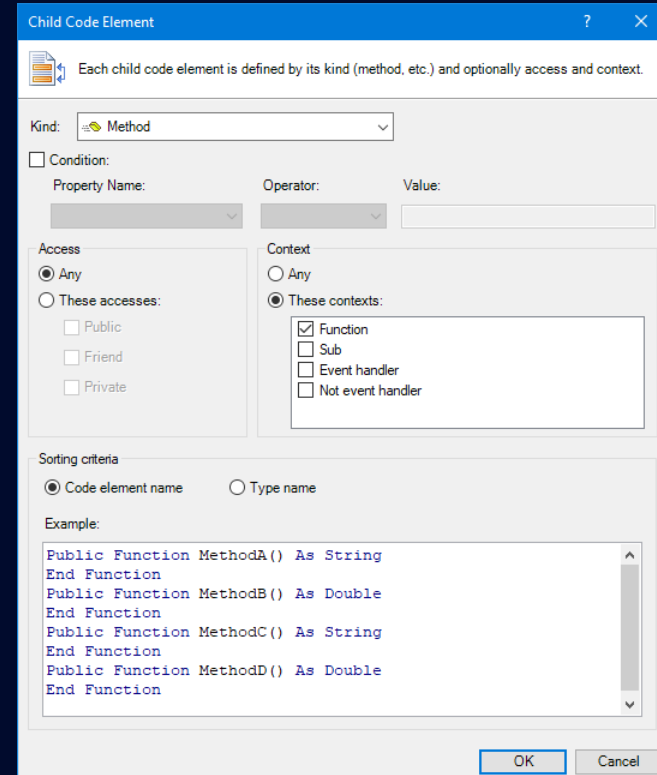
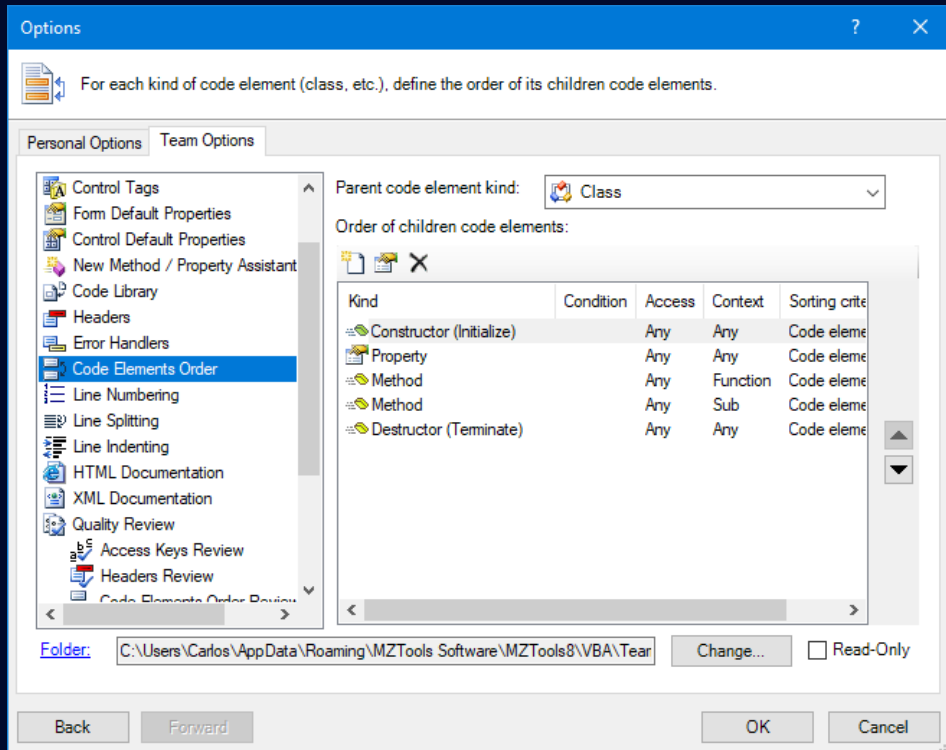
- More powerful than wildcards
- Captured groups and backreferences
- Positive / Negative Lookahead: find text only if some other text appears (or not) after that text
- Positive / Negative Lookbehind: find text only if some other text appears (or not) before that text

<https://www.regular-expressions.info/lookaround.html>



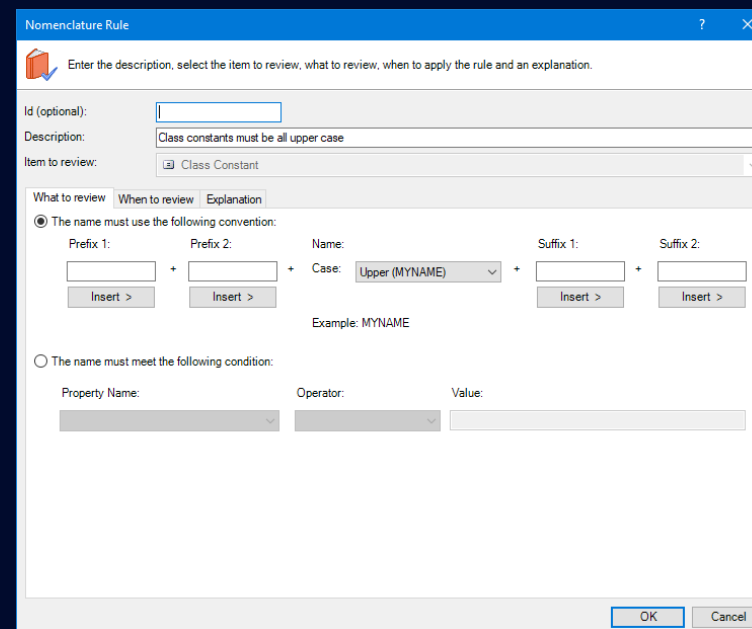
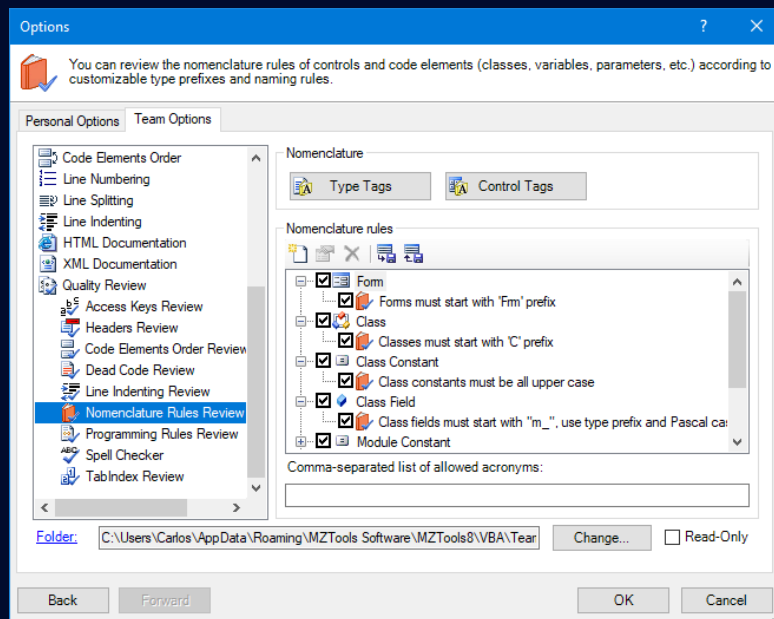
# 7. Custom code element sorting

- Sort methods of a file (class/module/interface) with 1-click using a custom criteria
- Allows automatic review of methods with incorrect order



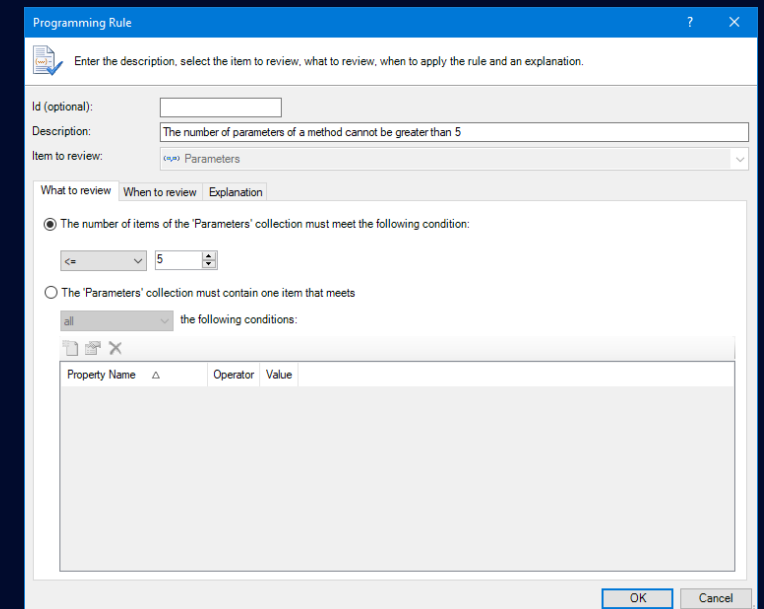
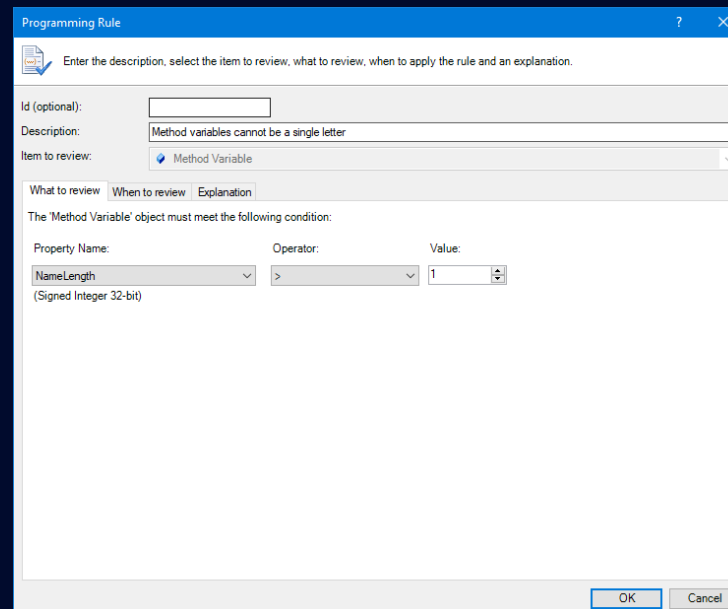
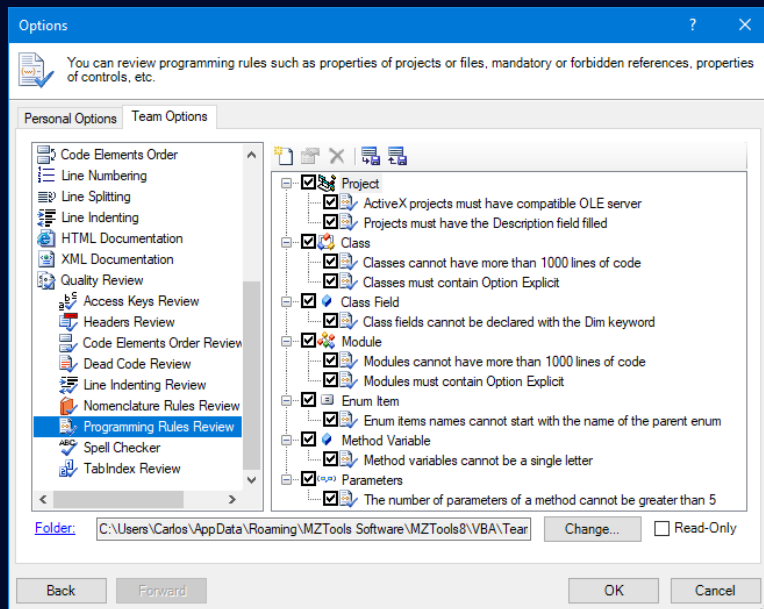
# 8. Nomenclature Rules Review

- Naming guidelines
- What To Review: Two options
  - Prefix 1 + Prefix 2 + name + Suffix 1+ Suffix 2
  - Name condition
- When To Review



# 9. Programming Rules Review

- Review anything about your code, files, references, etc.
- What To Review:
  - Individual item
  - Collections of items
- When To Review



# 10. Automating MZ-Tools features with scripts

- Execute script (VBScript, PowerShell, etc.) to automate MZ-Tools features with the command-line
- Example: before shipping:
  - Review Quality
  - Indent Lines
  - Add line numbers
  - Export Files (to add to source control)
  - Get Statistics (to store somewhere)
  - Generate XML/HTML Documentation (to publish somewhere)

```
Const officeProgId = "Access.Application"
Const addinProgId = "MZTools8VBA"
Const commandName = "MZTools8.AddLineNumbers"

Dim fso
Dim app
Dim addIn
Dim addInObject
Dim officeFile
Dim logFile
Dim commandLineArgs

' Set variables
officeFile = "C:\Users\Public\DatabaseAutomation.accdb"
logFile = "C:\Users\Public\MZTools8.log"
commandLineArgs = logFile

' Delete files if they exist
Set fso = CreateObject("Scripting.FileSystemObject")
If fso.FileExists(logFile) Then
    fso.DeleteFile(logFile)
End If

' Open the Office document
Set app = CreateObject(officeProgId)
app.OpenCurrentDatabase(officeFile)

' Make Office and VBA editor visible
app.Visible = True
app.VBE.MainWindow.Visible = True

' Get the MZ-Tools add-in
Set addIn = app.VBE.AddIns.Item(addinProgId)

' Ensure it is loaded
addIn.Connect = True

' Get its inner object
Set addInObject = addIn.Object

' Execute the command
Call addInObject.ExecuteCommand(commandName, commandLineArgs)

' Exit the Office application
app.Quit

WScript.Echo "Done"
```