**Настраиваем автоматическое закрытие Excel файла при отсутствии активности пользователя в течение определенного времени.**

Очень часто в повседневной деятельности нам приходится командой работать с Excel файлом, который находится в общем доступе на сетевом диске. В случае, если файл содержит умные таблицы, то предоставление общего доступа через Review->Share Workbook не представляется возможным из-за ограничений самого Excel. Это порождает проблему блокирования рабочего файла, если один забывчивый сотрудник оставил файл открытым после небольших коррекций и ушёл, например, обедать, а второму сотруднику срочно нужно внести в файл новую информацию. Примеров Вы сами можете вспомнить огромное количество.

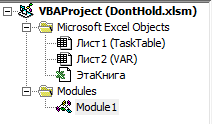
Ну что же, приступим!

Задача: Необходимо обеспечить автоматическое сохранение и закрытие файла после бездействия пользователя. При этом пользователь может вносить коррекции в файл, а потом бездействовать, либо просто открыть файл и ничего не делать. Под действием мы понимаем изменение любого диапазона ячеек (удаление значений, ввод новых данных и т.д.) на конкретном листе книги.

Решение:

В целом, чтобы выполнить все условия задачи необходимо будет использовать метод .OnTime. Он запускает таймер на выполнение процедуры закрытия файла. При этом нужно будет прерывать действие запущенного таймера при внесении новых изменений, и запускать его заново.

Теперь конкретно. Для начала приведу код, а потом всё объясню.



|  |  |
| --- | --- |
| **Module1**  Sub **DontHold**()  Dim KillTime\_Current As Date  Dim KillTime\_Previous As Date  KillTime\_Previous = Sheets("VAR").Cells(1, 4)  Sheets("VAR").Cells(2, 4) = KillTime\_Previous  If Sheets("VAR").Cells(1, 1) = 1 Then  Application.OnTime KillTime\_Previous, Procedure:="WbClose\_Without\_Change", Schedule:=False  End If  KillTime\_Current = Now + TimeSerial(0, 5, 0)  Sheets("VAR").Cells(1, 4) = KillTime\_Current  KillTime\_Current = Sheets("VAR").Cells(1, 4)  If Sheets("VAR").Cells(1, 1) = 1 Then  Application.OnTime KillTime\_Current, "WbClose”  Else  Application.OnTime KillTime\_Previous, Procedure:="WbClose", Schedule:=False  Application.OnTime KillTime\_Current, "WbClose"  End If  End Sub  Sub **WbClose**()  ThisWorkbook.Save  ThisWorkbook.Close  End Sub  Sub **WbClose\_Without\_Change**()  If Sheets("VAR").Cells(1, 1) = 0 Then  ThisWorkbook.Save  ThisWorkbook.Close  End If  End Sub | **Модуль листа TaskTable** (*рабочий лист для команды*)  Private Sub **Worksheet\_Change**(ByVal Target As Range)  If ThisWorkbook.ReadOnly Then  Else  Sheets("VAR").Cells(1, 1) = Sheets("VAR").Cells(1, 1) + 1  DontHold  End If  End Sub  **Модуль ЭтаКнига**  Private Sub **Workbook\_Open**()  If ThisWorkbook.ReadOnly Then  Else  Sheets("VAR").Cells(1, 1) = 0  Sheets("VAR").Cells(1, 4) = 0  Sheets("VAR").Cells(2, 4) = 0  Dim KillTime\_Current As Date  KillTime\_Current = Now + TimeSerial(0, 10, 0)  Sheets("VAR").Cells(1, 4) = KillTime\_Current  KillTime\_Current = Sheets("VAR").Cells(1, 4)  Application.OnTime KillTime\_Current, "WbClose\_Without\_Change"  End If  End Sub  Private Sub **Workbook\_BeforeClose**(Cancel As Boolean)  If ThisWorkbook.ReadOnly Then  Else  Dim KillTime\_Current As Date  KillTime\_Current = Sheets("VAR").Cells(1, 4)  If Sheets("VAR").Cells(1, 1) <> 0 Then  Application.OnTime KillTime\_Current, Procedure:="WbClose", Schedule:=False  ThisWorkbook.Save  ThisWorkbook.Close  Else  Application.OnTime KillTime\_Current, Procedure:="WbClose\_Without\_Change", Schedule:=False  ThisWorkbook.Save  ThisWorkbook.Close  End If  End If  End Sub  **Модуль листа VAR**  У него нет кода, это лист для хранения переменных. О нём расскажу по ходу пьесы. Можно скрыть в книге, чтобы никого не смущать. |

**Модуль листа TaskTable** (*рабочий лист для команды*)

‘процедура, срабатывающая при внесении изменений на лист

‘увеличивает счетчик изменений на 1

‘запускает процедуру DontHold, о ней ниже

‘если файл открыт только для чтения, то автоматическое закрытие файла происходить не будет.

Private Sub **Worksheet\_Change**(ByVal Target As Range)

If ThisWorkbook.ReadOnly Then

Else

Sheets("VAR").Cells(1, 1) = Sheets("VAR").Cells(1, 1) + 1

DontHold

End If

End Sub

**Module1**

‘процедура, которая будет запускаться, как только на рабочем листе что-то поменяется.

Sub **DontHold**()

‘переменная, содержащая в себе момент времени, в который произошло изменение

Dim KillTime\_Current As Date

‘если изменений становится больше одного, то время предыдущего измерения падает в переменную \_KillTime\_Previous. Нужно, чтобы убить предыдущий счетчик

Dim KillTime\_Previous As Date

‘лист VAR будет содержать в себе следующие ячейки с переменными: A1 – количество произведенных изменений с открытия файла, D1 – время текущего изменения, D2 – время предыдущего изменения.

См. комментарии для ЭтаКнига, процедура на событие открытия файла.

‘каждое изменение будет увеличивать A1 листа VAR на единицу.

‘для более стабильной работы макроса я решил перезаписывать результат функции “Now + TimeSerial(…)”, которое определяет моменты времени, сначала в лист VAR, а потом обратно в переменную. В противном случае Now будет меняться при повторном обращении к ней в команде на остановку счетчика. Подобные перезаписи будут встречаться по всему коду.

‘считываем время предыдущего изменения

KillTime\_Previous = Sheets("VAR").Cells(1, 4)

Sheets("VAR").Cells(2, 4) = KillTime\_Previous

‘с первым изменением мы останавливаем счетчик на закрытие книги, который запускается при её открытии

If Sheets("VAR").Cells(1, 1) = 1 Then

Application.OnTime KillTime\_Previous, Procedure:="WbClose\_Without\_Change", Schedule:=False

End If

‘считываем время текущего изменения

‘время на бездействие после последнего изменения можете варьировать TimeSerial(часы, минуты, секунды)

KillTime\_Current = Now + TimeSerial(0, 5, 0)

Sheets("VAR").Cells(1, 4) = KillTime\_Current

KillTime\_Current = Sheets("VAR").Cells(1, 4)

‘если это первое изменение, то не делаем остановку предыдущего счетчика, так как ранее мы сделали его остановку. (для первого изменения предыдущим как бы было открытие книги). Запускаем новый счетчик на закрытие.

If Sheets("VAR").Cells(1, 1) = 1 Then

Application.OnTime KillTime\_Current, "WbClose”

Else

‘для второго и следующих событий (A1>1) убиваем предыдущий счетчик и запускаем новый, используя соответствующие переменные

Application.OnTime KillTime\_Previous, Procedure:="WbClose", Schedule:=False

Application.OnTime KillTime\_Current, "WbClose"

End If

End Sub

‘процедура закрытия простенькая, сохраняем и закрываем книгу

Sub **WbClose**()

ThisWorkbook.Save

ThisWorkbook.Close

End Sub

‘процедура закрытия для случая, если в листе не было изменений

Sub **WbClose\_Without\_Change**()

If Sheets("VAR").Cells(1, 1) = 0 Then

ThisWorkbook.Save

ThisWorkbook.Close

End If

End Sub

**Модуль ЭтаКнига**

‘процедура, срабатывающая на открытие книги

Private Sub **Workbook\_Open**()

‘если книга открыта в режиме только чтение то ничего происходить не будет

If ThisWorkbook.ReadOnly Then

Else

‘обнуляем переменные на листе VAR для начала

Sheets("VAR").Cells(1, 1) = 0

Sheets("VAR").Cells(1, 4) = 0

Sheets("VAR").Cells(2, 4) = 0

Dim KillTime\_Current As Date

‘время на бездействие после открытия книги можете варьировать TimeSerial(часы, минуты, секунды)

KillTime\_Current = Now + TimeSerial(0, 10, 0)

Sheets("VAR").Cells(1, 4) = KillTime\_Current

KillTime\_Current = Sheets("VAR").Cells(1, 4)

‘запускаем таймер на закрытие книги

Application.OnTime KillTime\_Current, "WbClose\_Without\_Change"

End If

End Sub

‘очень важно при закрытии книги убить все запущенные счетчики. Иначе файл после закрытия будет самопроизвольно открываться, чтобы завершить выполнение процедуры и выполнить своё предназначение. Для этого сделаем событие закрытие книги:

Private Sub **Workbook\_BeforeClose**(Cancel As Boolean)

‘если книга открыта в режиме только чтение то ничего происходить не будет

If ThisWorkbook.ReadOnly Then

Else

Dim KillTime\_Current As Date

KillTime\_Current = Sheets("VAR").Cells(1, 4)

‘если изменения в книге были, то нам нужно остановить последний открытый счетчик

If Sheets("VAR").Cells(1, 1) <> 0 Then

Application.OnTime KillTime\_Current, Procedure:="WbClose", Schedule:=False

ThisWorkbook.Save

ThisWorkbook.Close

Else

‘ в противном случае останавливаем открытый счетчик, который был запущен при открытии книги

Application.OnTime KillTime\_Current, Procedure:="WbClose\_Without\_Change", Schedule:=False

ThisWorkbook.Save

ThisWorkbook.Close

End If

End If

End Sub

‘Если вы будете модифицировать макрос, то очень важно придерживаться следующего принципа касательно отключения счетчиков.

‘Для отключения счетчика у вас должен быть открытый счетчик с точно таким же временем(название переменной не важно, важно значение) и на точно такую же процедуру. В противном случае Excel будет осеивать вас ошибками по этому поводу.