**Функции, используемые в процедурах VBA**

Настоящая заметка продолжает знакомство с VBA. В ней представлены некоторые «практичные» функции, которые могут использоваться в ваших приложениях либо помогут в создании аналогичных функций. Эти функции наиболее полезны, когда вызываются из другой процедуры VBA. Следовательно, они объявляются с ключевым словом Private и не отображаются в диалоговом окне Excel *Мастер функций* (подробнее см. [Работа с процедурами VBA](http://baguzin.ru/wp/?p=16451)).[[1]](#footnote-1)

### Функция FileExists

Данная функция получает один аргумент (путь и имя файла) и возвращает ИСТИНА, если файл существует.

Private Function FileExists(fname) As Boolean

' Возвращает TRUE, если файл существует

Dim x As String

x = Dir(fname)

If x <> "" Then FileExists = True \_

Else FileExists = False

End Function

### Функция FileNameOnly

Функция получает один аргумент (путь и имя файла; не забывайте брать строку в кавычки) и возвращает только имя файла.

Private Function FileNameOnly(pname) As String

' Возвращает имя файла из строки путь/имя файла

Dim temp As Variant

temp = Split(pname, Application.PathSeparator)

FileNameOnly = temp(UBound(temp))

End Function

Функция использует функцию VBA Split, которая принимает строку (вместе с символами-разделителями) и возвращает массив типа variant, содержащий элементы, которые находятся между символами-разделителями. В рассматриваемом случае переменной temp присваивается массив, содержащий текстовые строки между Application.PathSeparater (обычно в качестве разделителя используется обратная косая черта).

Если в качестве аргумента указать "с:\excelfiles\2010\backup\budget.xlsx", функция возвратит строку budget.xlsx.

Функция FileNameOnly обрабатывает любой путь и имя файла (даже если файла не существует). Если файл существует, лучше воспользоваться следующей более простой функцией.

Private Function FileNameOnly2(pname) As String

FileNameOnly2 = Dir(pname)

End Function

### Функция PathExists

Функция получает один аргумент (путь) и возвращает ИСТИНА, если путь существует.

Private Function PathExists(pname) As Boolean

' Возвращает TRUE, если путь существует

If Dir(pname, vbDirectory) = "" Then

PathExists = False

Else

PathExists = (GetAttr(pname) And vbDirectory) = vbDirectory

End If

End Function

### Функция RangeNameExists

Функция получает один аргумент (название диапазона) и возвращает ИСТИНА, если в активной рабочей книге существует указанное название диапазона.

Private Function RangeNameExists(nname) As Boolean

' Возвращает TRUE, если имя диапазона существует

Dim n As Name

RangeNameExists = False

For Each n In ActiveWorkbook.Names

If UCase(n.Name) = UCase(nname) Then

RangeNameExists = True

Exit Function

End If

Next n

End Function

### Функция SheetExists

Функция получает один аргумент (название рабочего листа) и возвращает ИСТИНА, если данный рабочий лист существует в активной рабочей книге.

Private Function SheetExists(sname) As Boolean

' Возвращает TRUE, если лист существует в активной рабочей книге

Dim x As Object

On Error Resume Next

Set x = ActiveWorkbook.Sheets(sname)

If Err = 0 Then SheetExists = True \_

Else SheetExists = False

End Function

### Функция WorkbooklsOpen

Функция получает один аргумент (название рабочей книги) и возвращает ИСТИНА, если данная рабочая книга открыта.

Private Function WorkbookIsOpen(wbname) As Boolean

' Возвращает TRUE, если рабочая книга открыта

Dim x As Workbook

On Error Resume Next

Set x = Workbooks(wbname)

If Err = 0 Then WorkbookIsOpen = True \_

Else WorkbookIsOpen = False

End Function

### Проверка принадлежности к коллекции

Следующая функция представляет собой образец «групповой» функции, с помощью которой можно определить, является ли объект членом коллекции.

Private Function IsInCollection(Coin As Object, \_

Item As String) As Boolean

Dim Obj As Object

On Error Resume Next

Set Obj = Coin(Item)

IsInCollection = Not Obj Is Nothing

End Function

Эта функция имеет два аргумента: коллекцию (объект) и элемент (строка), который может быть либо не быть членом коллекции. Функция будет создавать объектную переменную, представляющую элемент коллекции. Если попытка увенчается успехом, функция возвратит True; иначе — False. Функцию IsInCollection можно использовать вместо трех других функций, приведенных выше. Чтобы определить, содержится ли в активной рабочей книге диапазон Data, вызовите функцию IsInCollection с помощью следующего оператора:

MsgBox IsInCollection(ActiveWorkbook.Names, "Data")

Для того чтобы определить, открыта ли рабочая книга с названием Budget, используйте следующий оператор:

MsgBox IsInCollection(Workbooks, "budget.xlsx")

Чтобы узнать, содержит ли активная рабочая книга рабочий лист Лист1, используйте следующий оператор:

MsgBox IsInCollection(ActiveWorkbook.Worksheets, "Лист1")

### Получение значения из закрытой рабочей книги

В VBA не существует метода получения значения из закрытого файла рабочей книги. Однако вы можете воспользоваться возможностью управления ссылками на файлы, которая предоставляется в Excel. В настоящем разделе описана функция VBA GetValue, которая получает значение из закрытой книги. Эта задача выполняется в результате вызова макроса XLM, который появился в «доисторических» версиях Excel (до версии 5), но поддерживается до сих пор (подробнее о макрофункциях xlm см. [Функция Получить.Ячейку](http://baguzin.ru/wp/?p=12814)).

Private Function GetValue(path, file, sheet, ref)

' Выборка значения из закрытой книги

Dim arg As String

' Проверка существования файла

If Right(path, 1) <> "\" Then path = path & "\"

If Dir(path & file) = "" Then

GetValue = "Файл не найден"

Exit Function

End If

' Создание аргумента

arg = "'" & path & "[" & file & "]" & sheet & "'!" & \_

Range(ref).Range("A1").Address(, , xlR1C1)

' Вызов макроса XLM

GetValue = ExecuteExcel4Macro(arg)

End Function

Функция GetValue принимает четыре аргумента:

* path — путь к закрытому файлу (например, "d:\files");
* file — название рабочей книги (например, "budget.xlsх");
* sheet — название рабочего листа (например, "Лист1");
* ref — ссылка на ячейку (например, "С4").

Следующая процедура демонстрирует, как используется функция GetValue. В этой процедуре отображается значение ячейки С1 листа Лист2 файла Закрытая\_книга.xlsx, расположенного в текущей папке:

Sub TestGetValue()

Dim p As String, f As String

Dim s As String, a As String

p = ThisWorkbook.path

f = "Закрытая\_книга.xlsx"

s = "Лист2"

a = "C1"

MsgBox GetValue(p, f, s, a)

End Sub

Ниже приведен еще один пример. Эта процедура считывает 160 значений (20 строк и 8 столбцов) из закрытого файла и помещает эти значения на активный рабочий лист:

Sub TestGetValue2()

Dim p As String, f As String

Dim s As String, a As String

Dim r As Long, c As Long

p = ThisWorkbook.path

f = "Закрытая\_книга.xlsx"

s = "Лист1"

Application.ScreenUpdating = False

For r = 1 To 20

For c = 1 To 8

a = Cells(r, c).Address

Cells(r, c) = GetValue(p, f, s, a)

Next c

Next r

End Sub

Функция GetValue не работает, если ее использовать в формуле рабочего листа. Эту функцию вообще не следует использовать в листах Excel, поскольку для получения значения из закрытого файла можно просто создать формулу со ссылкой.

1. По материалам книги [Джон Уокенбах. Excel 2010. Профессиональное программирование на VBA](http://baguzin.ru/wp/?p=16255). – М: Диалектика, 2013. – С. 358–362. [↑](#footnote-ref-1)