**Вызов функций Windows API из кода VBA**

Одна из интересных возможностей VBA — поддержка функций, которые хранятся в динамически подключаемых библиотеках (Dynamic Link Libraries – DLL). В заметке демонстрируются популярные функции Windows API.[[1]](#footnote-1) Для простоты изложения представленные здесь объявления API-функций могут корректно выполняться только в среде Excel 2010 и более поздних версиях. В то же время, примеры файлов совместимы с предыдущими версиями Excel.



Рис. 1. Выбор файла для поиска приложения

### Определение связей с файлами

В Windows многие типы файлов ассоциируются с конкретным приложением. Эта связь позволяет загрузить файл в соответствующее приложение (для этого дважды щелкните мышью на файле). Функция GetExecutable вызывает функцию Windows API с целью получить полный путь к приложению, связанному с указанным файлом. Например, в системе находится ряд файлов с расширением .txt; вероятно, один такой файл с названием Readme.txt в данный момент расположен в папке Windows. Функцию GetExecutable можно применять для определения полного пути приложения (которое запускается после двойного щелчка на выбранном файле).

Private Declare Function FindExecutableA Lib "shell32.dll" \_

 (ByVal lpFile As String, ByVal lpDirectory As String, \_

 ByVal lpResult As String) As Long

Function GetExecutable(strFile As String) As String

 Dim strPath As String

 Dim intLen As Integer

 strPath = Space(255)

 intLen = FindExecutableA(strFile, "\", strPath)

 GetExecutable = Trim(strPath)

End Function

Откройте файл *file association.xlsm*, кликните на кнопке «Определение связей между файлами», в открывшемся окне выберите файл (см. рис. 1), кликните *Открыть*. Функция GetExecutable вернет полный путь к приложению, которое связано с выбранным файлом (рис. 2).



Рис. 2. Определение пути и имени для приложения, связанного с заданным файлом

### Определение буквы диска

В VBA нет способа получения информации о дисковых накопителях. Но эта проблема легко решается с помощью трех API-функций, обеспечивающих получение всех необходимых сведений. Откройте файл *drive information.xlsm*, кликните на копку. Процедуры VBA идентифицирует все подключенные дисковые накопители, определяет их тип, а также указывает размер свободного и занятого пространства на диске (рис. 3). Код используемых в примере функций Windows API можно найти в модуле Excel-файла.



Рис. 3. С помощью функций Windows API можно получить всю информацию о дисках, установленных в системе

### Определение параметров принтера по умолчанию

Функция Windows API может быть использована для получения информации об активном принтере. Данная информация содержится в одной текстовой строке. Программа анализирует эту строку и отображает информацию в более удобном для чтения формате. Откройте файл *printer info.xlsm*, нажмите на кнопку. Свойство ActivePrinter объекта Application возвращает название активного принтера (и позволяет его изменить). Но не существует способа определить используемый драйвер принтера или порт. Поэтому функция DefaultPrinterInfo() столь полезна. После выполнения процедуры на экран выводится окно сообщения (рис. 4).



Рис. 4. Получение информации об активном принтере с помощью функции Windows API

### Определение текущего видеорежима

Если в приложении необходимо отобразить определенный объем информации на одном экране, то, зная размер экрана, можно правильно задать масштаб текста. Кроме того, в коде определяется количество мониторов в системе. Если установлено более одного монитора, процедура определяет размер виртуального экрана. Откройте файл *video mode.xlsm*, кликните на кнопке, и процедура вернет видеорежим (рис. 5).



Рис. 5. Использование функций Windows API для определения видеорежима монитора

Option Explicit

#If VBA7 And Win64 Then

 Declare PtrSafe Function GetSystemMetrics Lib "user32" \_

 (ByVal nIndex As Long) As Long

#Else

Declare Function GetSystemMetrics Lib "user32" \_

 (ByVal nIndex As Long) As Long

#End If

Public Const SM\_CMONITORS = 80

Public Const SM\_CXSCREEN = 0

Public Const SM\_CYSCREEN = 1

Public Const SM\_CXVIRTUALSCREEN = 78

Public Const SM\_CYVIRTUALSCREEN = 79

Sub DisplayVideoInfo()

 Dim numMonitors As Long

 Dim vidWidth As Long, vidHeight As Long

 Dim virtWidth As Long, virtHeight As Long

 Dim Msg As String

 numMonitors = GetSystemMetrics(SM\_CMONITORS)

 vidWidth = GetSystemMetrics(SM\_CXSCREEN)

 vidHeight = GetSystemMetrics(SM\_CYSCREEN)

 virtWidth = GetSystemMetrics(SM\_CXVIRTUALSCREEN)

 virtHeight = GetSystemMetrics(SM\_CYVIRTUALSCREEN)

 If numMonitors > 1 Then

 Msg = numMonitors & " мониторов" & vbCrLf

 Msg = Msg & "Виртуальный экран: " & virtWidth & " X "

 Msg = Msg & virtHeight & vbCrLf & vbCrLf

 Msg = Msg & "Видеорежим основного монитора: "

 Msg = Msg & vidWidth & " X " & vidHeight

 Else

 Msg = Msg & "Видеорежим монитора: "

 Msg = Msg & vidWidth & " X " & vidHeight

 End If

 MsgBox Msg

End Sub

### Добавление звука в приложение

Можно расширить возможности Excel по воспроизведению звука в форматах WAV и MIDI. Откройте файл *sound.xlsm*, находящийся в папке sound, и измените значение в одной из ячеек диапазона В4:В12. Как только сумма в ячейке В13 достигнет значения 1000, прозвучит сигнал (рис. 6). Функция Alarm предназначена для применения в формуле рабочего листа. Она использует функцию Windows API для проигрывания звука, если ячейка соответствует определенному условию.



Рис. 6. Если сумма в ячейке В13 достигнет 1000, прозвучит сигнал

Function ALARM(Cell, Condition)

 Dim WAVFile As String

 Const SND\_ASYNC = &H1

 Const SND\_FILENAME = &H20000

 If Evaluate(Cell.Value & Condition) Then

 WAVFile = ThisWorkbook.Path & "\sound.wav"

 Call PlaySound(WAVFile, 0&, SND\_ASYNC Or SND\_FILENAME)

 ALARM = True

 Else

 ALARM = False

 End If

End Function

Функция Alarm имеет два аргумента: ссылку на ячейку и условие (выраженное в виде строки). Формула =ALARM (В13, ">=1000") использует функцию Alarm для проигрывания WAV-файла, если значение в ячейке В13 больше или равно 1000. Функция использует функцию VBA Evaluate для определения, соответствует ли значение ячейки заданному критерию. Если условие выполнено (и звук воспроизведен), функция возвращает ИСТИНА, в противном случае она возвращает значение ЛОЖЬ.

### Чтение и запись параметров системного реестра

Многие приложения Windows используют системный реестр для хранения параметров. Процедуры VBA могут считывать значения из реестра и записывать в него новые значения. Функции VBA GetRegistry и WriteRegistry – две функции-«оболочки», упрощающие управление реестром. Откройте файл *windows registry.xlsm* и изучите код VBA.

Функция GetRegistry возвращает раздел из указанного места регистра. У нее три аргумента:

* *Root Key*. Строка, представляющая ветвь реестра, к которой обращается функция. Данная строка может принимать одно из следующих значений: HKEY\_CLASSES\_ROOT, HKEY\_CURRENT\_USER, HKEY\_LOCAL\_MACHINE, HKEY\_USERS, HKEY\_CURRENT\_CONFIG, HKEY\_DYN\_DATA.
* *Path*. Полный путь к разделу реестра, к которому обращается функция.
* *RegEntry*. Название параметра, который должна получить функция.

Например, если необходимо найти графический файл, используемый в качестве обоев рабочего стола, используйте функцию GetRegistry в следующей процедуре (обратите внимание, что аргументы не чувствительны к регистру).

Sub Wallpaper()

 Dim RootKey As String

 Dim Path As String

 Dim RegEntry As String

 RootKey = "hkey\_current\_user"

 Path = "Control Panel\Desktop"

 RegEntry = "Wallpaper"

 MsgBox GetRegistry(RootKey, Path, RegEntry), \_

vbInformation, Path & "\RegEntry"

End Sub

Напоминаю, чтобы вызвать процедуру пройдите по меню Вид –> Макросы –> Макросы, выделите процедуру *Wallpaper*, и кликните *Выполнить*. После выполнения этой процедуры в окне сообщения отображаются путь и имя графического файла (либо пустая строка, если обои не используются).

Функция WriteRegistry записывает значение в указанный раздел реестра. Если операция завершается успешно, функция возвращает ИСТИНА; в противном случае функция возвращает ЛОЖЬ. Первые три аргумента функция WriteRegistry такие же, как и аргументы GetRegistry, но также есть и четвертый аргумент – RegVal – значение, которое записывается в реестр. Например, процедура Workbook\_Open() записывает текущую дату и время в разделе настроек Excel.

Sub Workbook\_Open()

 RootKey = "hkey\_current\_user"

 Path = "software\microsoft\office\14.0\excel\LastStarted"

 RegEntry = "DateTime"

 RegVal = Now()

 If WriteRegistry(RootKey, Path, RegEntry, RegVal) Then

 Msg = RegVal & " сохранено в реестре."

 Else

 Msg = "произошла ошибка"

 End If

 MsgBox Msg

End Sub

Если вы решили воспользоваться системным реестром для хранения и выборки настроек приложений Excel, проще обратиться к функциям VBA GetSetting и SaveSetting. Важно понимать, что они работают только со следующим разделом реестра:

HKEY\_CURRENT\_USER\Software\VB and VBA Program Settings

Другими словами, с помощью этих функций можно управлять данными только одной ветви реестра, в которой сохраняются базовые настройки Excel.

1. По материалам книги [Джон Уокенбах. Excel 2010. Профессиональное программирование на VBA](http://baguzin.ru/wp/?p=16255). – М: Диалектика, 2013. – С. 376–383. [↑](#footnote-ref-1)