

Макрос (VBA) для создания отчета о воспроизведении в FDOнAir



Пример макроса

*Дата выпуска:
15 декабря 2016 г.*

Краткое руководство



Содержание

| | |
|--|----|
| Введение | 3 |
| Настройка программы FDO nAir | 4 |
| Создание макроса на языке Visual Basic в приложении Microsoft Excel | 5 |
| Подключение и использование макроса в приложении Microsoft Excel | 15 |



Введение

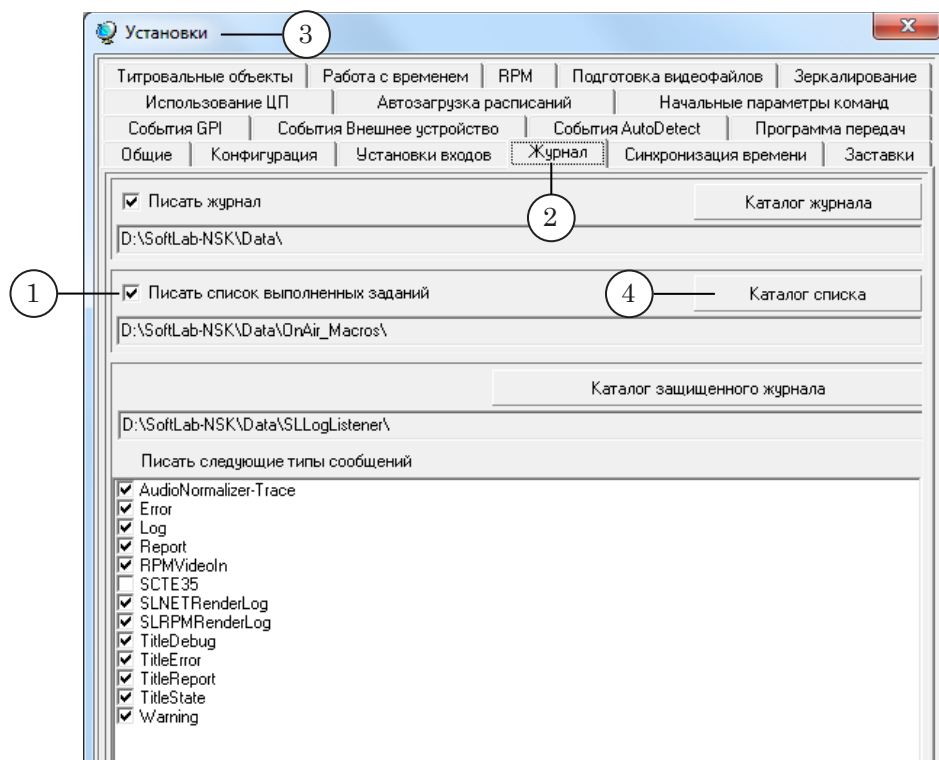
В данном документе описан пример того, как можно по журналу работы приложения FDOнAir составить отчёт о воспроизведённых в эфир видеоданных за определённый период времени. За основу берётся файл со списком выполненных заданий, созданный в процессе работы FDOнAir, и программа-макрос, написанная на языке Visual Basic для Приложений (VBA).

В следующих главах приведён текст макроса, описана последовательность действий по соответствующим настройкам работы FDOнAir, созданию и подключению макроса из Microsoft Excel (приложение из пакета Microsoft Office) и последующей работе с ним.



Настройка программы FDO n Air

Для составления отчёта о телевещании с помощью программы FDO n Air необходим список выполненных заданий. Такой список создается автоматически в процессе работы программы, если перед началом вещания включить опцию **Писать список выполненных заданий** (1) на вкладке **Журнал** (2) окна настроек **Установки** (3). Чтобы открыть это окно, нужно нажать кнопку **Установки** в главном окне приложения FDO n Air.



На этой же вкладке с помощью кнопки **Каталог списка** (4) необходимо выбрать папку, в которую будет записываться файл со списком выполненных команд. Имя файла, организованного в формате XML, формируется как `airMMДД.PlayReport`, где `MM` – текущий месяц, а `ДД` – день работы программы. При работе приложения FDO n Air в этот файл сохраняются записи о названиях воспроизведённых файлов, дате и времени начала воспроизведения и другая информация. Более подробное описание структуры файла `airMMДД.PlayReport` и типов команд, записываемых в него, можно найти в руководстве пользователя [«FDO n Air. Автоматизация вещания»](#), разделе «7.1. Структура файла *.PlayReport».

Примечание: Подробная информация о работе с журналами FDO n Air содержится в руководстве [«Лог-файлы FDO n Air. Настройка записи и просмотр»](#).

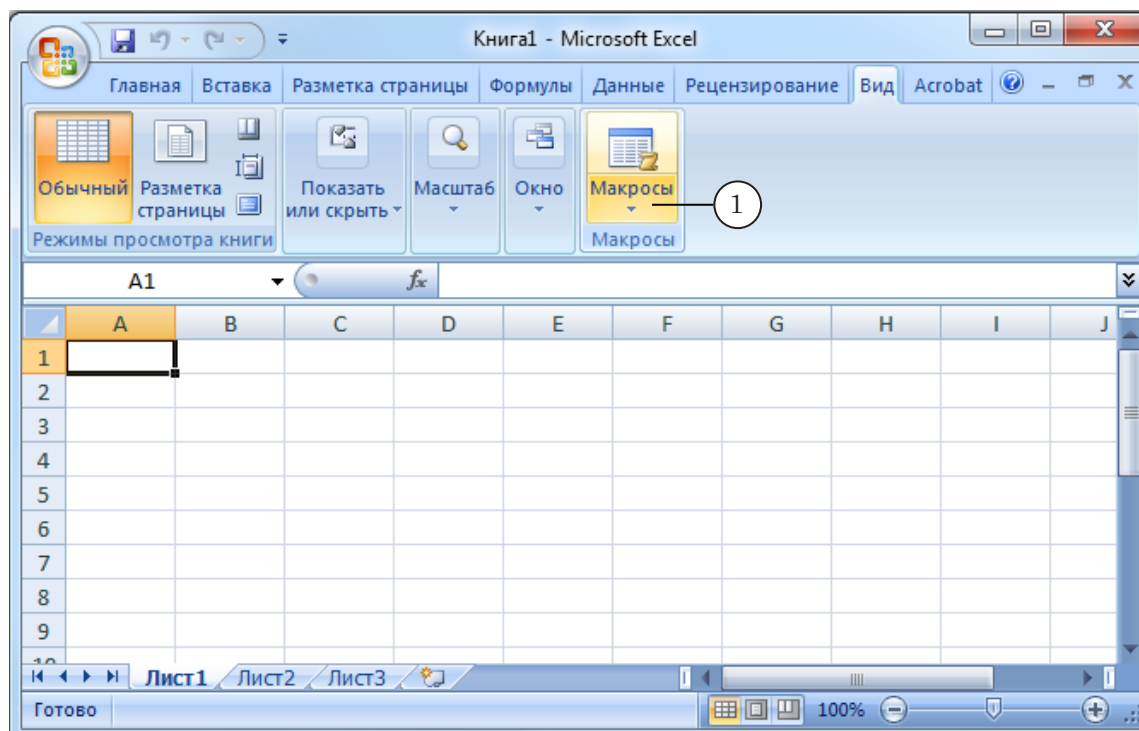


Создание макроса на языке Visual Basic в приложении Microsoft Excel

Чтобы составить отчёт о телевещании за определённый интервал времени, в списке выполненных заданий из отчёта airMMDD.PlayReport нужно выбрать файлы заданного типа. Например, это могут быть только фильмы – файлы с расширением *.avi; *.html; *.mpg; *.mpeg; *.mpv; *.m2v; *.m2p; *.mp2, или только статичные изображения – файлы с расширением *.tga;*.bmp;*.dib;*.jpg;*.png;*.422. В данном разделе приведен пример макроса на языке Visual Basic, который позволит автоматизировать процедуру выбора только фильмов из списка воспроизведённых в FDO nAir заданий, и показаны шаги его создания в приложении Microsoft Excel.

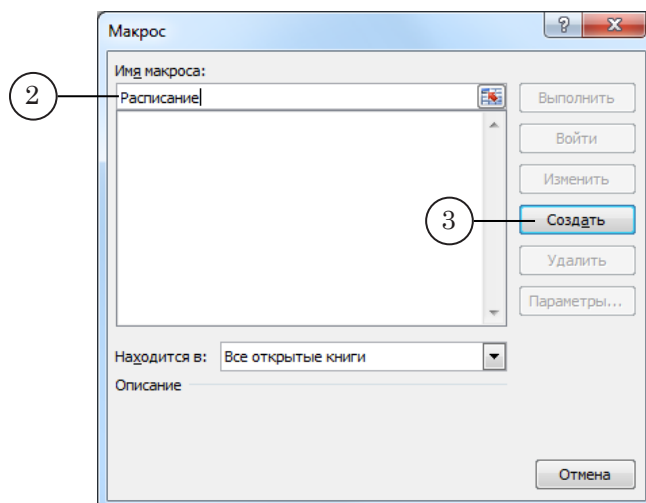
Примечание: Если вы решите использовать предлагаемый пример в качестве образца, то при копировании текста макроса в окно редактора Microsoft Visual Basic убедитесь, что копируемый блок текста не содержит лишние элементы (например, номера страниц руководства и пр.)

1. Запустите приложение Microsoft Excel.
2. Для создания макроса из меню Вид приложения Microsoft Excel вызовите диалог Макросы (1).

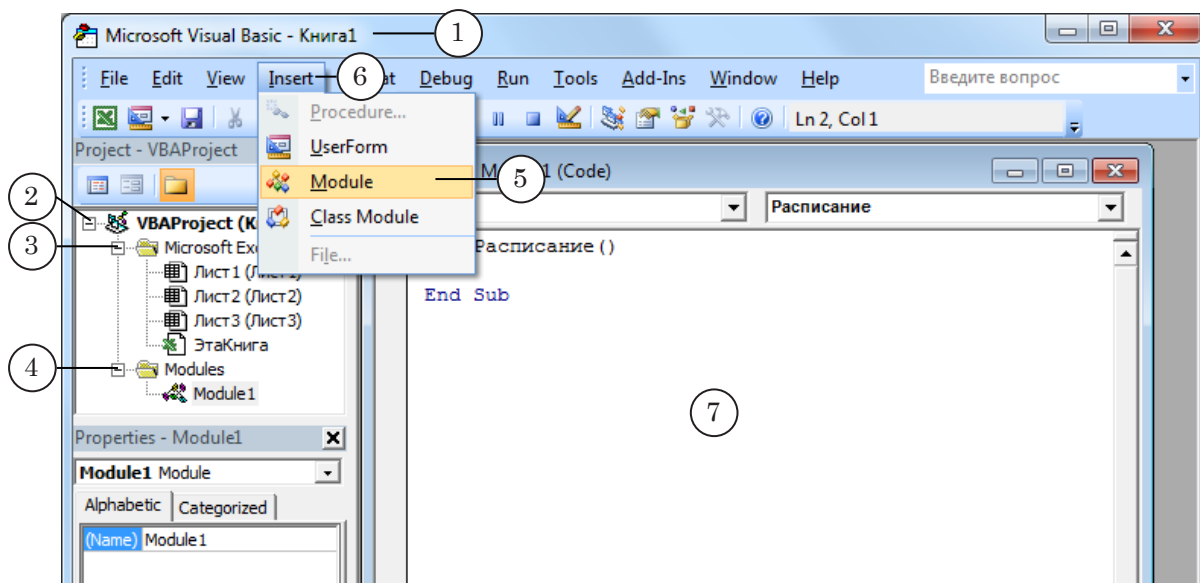




3. В поле Имя макроса (2) задайте имя макроса, а затем выполните команду Создать (3).



4. Откроется окно редактора Microsoft Visual Basic (1). В окне Project – VBAProject отображается схема проекта созданного макроса VBAProject (Книга1) (2).



Проект создаваемого макроса VBAProject по умолчанию включает в себя 2 раздела:

- Microsoft Excel Objects (3) с подразделами:
 - Лист 1 (Лист 1);
 - Лист 2 (Лист 2);
 - Лист 3 (Лист 3);
 - ЭтаКнига – в нашем примере в этот подраздел будет добавлена часть макроса, которая отвечает за инициализацию кнопки Import OnAir PlayReport-file (см. ниже п. 5);
- Modules (4) с одним подразделом Module1. Если раздел Modules или включённый в него подраздел Module1 отсутствуют, их можно добавить командой



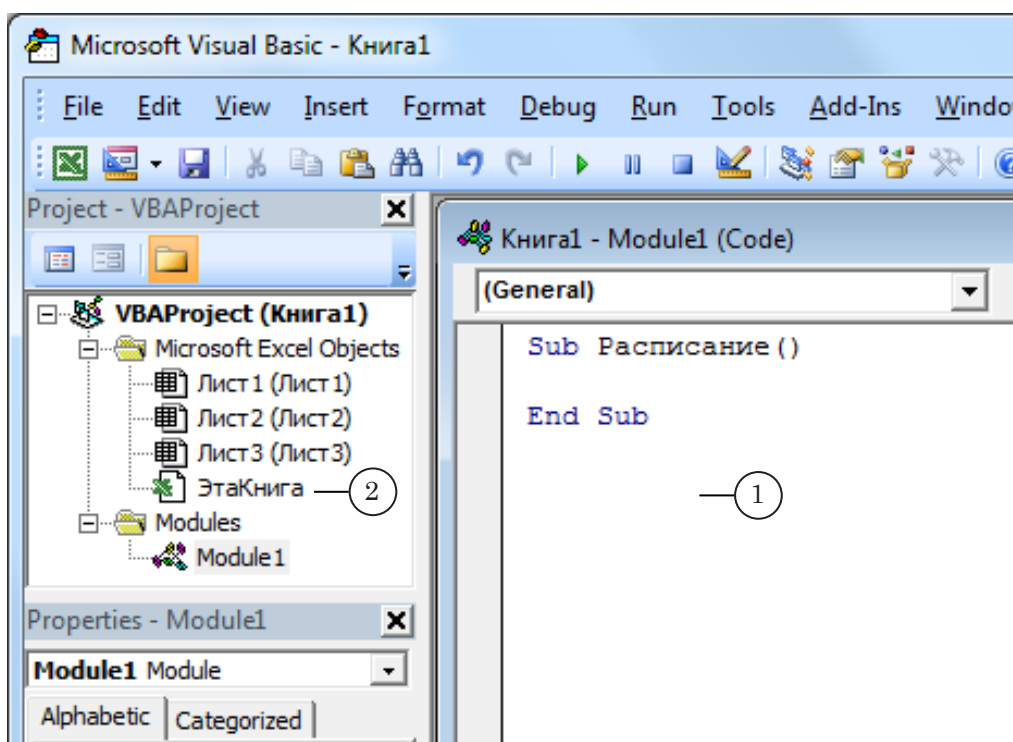
Module (5) из меню Insert (6) редактора. В нашем примере потребуется добавить два подраздела в раздел Module – для модулей ScanOnAirPlayReportCommandBar и ScanOnAirPlayReportXml (см. ниже п. 5).

Чтобы открыть окно (7) для редактирования содержимого какого-либо подраздела, дважды щелкните по его заголовку в схеме проекта.

5. Скопируйте текст функций и процедур на языке Visual Basic, приведённый ниже, в соответствующие подразделы (и при необходимости отредактируйте).

1. Часть, которая отвечает за инициализацию кнопки Import OnAir PlayReport-file при включении макроса и за удаление её после его отключения.

Эта часть включает в себя команды, инициализирующие создание кнопки Import OnAir PlayReport-file на панели рабочей области окна приложения Microsoft Excel. Кнопка позволяет вызвать диалог выбора файла заданий, выполненных программой FDonAir – airММДД.PlayReport.



Поскольку кнопка Import OnAir PlayReport-file должна быть выведена именно на панели рабочей области, эта часть макроса должна быть связана непосредственно с рабочей областью приложения. Поэтому текст, приведённый ниже, следует скопировать в поле (1) подраздела ЭтаКнига (2) раздела Microsoft Excel Objects.

Примечание: Дважды щелкните по заголовку, чтобы открыть окно редактирования.



```
Option Explicit
Const PlugInName As String = "Import OnAir PlayReport-file"

Private Sub Workbook_AddinInstall()
    ScanOnAirPlayReportCommandBar.InstallCommandBar (PlugInName)
End Sub

Private Sub Workbook_AddinUninstall()
    ScanOnAirPlayReportCommandBar.RemoveCommandBar (PlugInName)
End Sub
```

2. Исполнимый модуль ScanOnAirPlayReportCommandBar. Модуль отвечает за создание кнопки Import OnAir PlayReport-file. Текст, приведенный ниже, следует скопировать в поле подраздела раздела Modules, названного ScanOnAirPlayReportCommandBar (про добавление подразделов см. пункт 4).

```
Option Explicit
Function InstallCommandBar(Name As String) As CommandBar

    Dim NewBtn As CommandBarControl

    On Error GoTo ErrHandler

    Set InstallCommandBar = Application.CommandBars.Add(Name,
        msoBarTop, False, False)

    Set NewBtn = InstallCommandBar.Controls.Add(msoControlButton)
    NewBtn.Style = msoButtonCaption
    NewBtn.Caption = "Import OnAir PlayReport-file"
    NewBtn.OnAction = "ScanOnAirPlayReportXml.MakeAirXls"

    InstallCommandBar.Visible = True

    Exit Function
ErrHandler:

    MsgBox Err.Description

End Function
Sub RemoveCommandBar(Name As String)
    On Error GoTo ErrHandler
    Application.CommandBars(Name).Delete
```




Exit Sub

ErrorHandler:

MsgBox Err.Description

End Sub

3. Исполнимый модуль ScanOnAirPlayReportXml.

Функции и процедуры модуля ScanOnAirPlayReportXml осуществляют работу с выбранным файлом airMMДД.PlayReport. Текст, приведенный ниже, следует скопировать в поле подраздела раздела Modules, названного ScanOnAirPlayReportXml (про добавление подразделов см. пункт 4).

‘ Процедура управления, запускает последовательно процедуры
‘ для открытия файла и заполнения таблицы

```
Public Sub MakeAirXls()  
    Dim StrFileName As String  
    StrFileName = GetFileName  
    If StrFileName <> "False" Then  
        CreateTable  
        FillTableFromFile (StrFileName)  
    Else  
        MsgBox ("File is not selected")  
    End If  
End Sub
```

‘ Функция вызова диалога выбора файла
‘ возвращает имя выбранного файла

```
Public Function GetFileName() As String  
    Dim FilterFile As String  
    Dim FileName As String  
    FilterFile = "Schedule (*.PlayReport),*.PlayReport," & _  
        "All files (*.*)*.*,*.*,"  
    FileName = Application.GetOpenFilename(FilterFile, 1)  
    GetFileName = FileName  
End Function
```

‘ Функция создания таблицы, задаёт имя листа и колонок,
‘ формат колонки "Time",
‘ в которую можно записывать только целое число

```
Public Function CreateTable()  
    Sheets(1).Name = "Playback-list"  
    Sheets("Playback-list").Columns("A:C").Clear  
    With Sheets("Playback-list")
```



```
.Cells(1, 1) = "Date"  
.Cells(1, 2) = "Time"  
.Cells(1, 3) = "File"  
.Columns(2).NumberFormat = "00\00\00\00"  
End With  
End Function
```

```
‘ Функция, которая открывает файл, считывает с него данные  
‘ и заполняет колонки таблицы  
Public Function FillTableFromFile(FileName As String)
```

```
Const PosDate As Integer = 1  
Const PosTime As Integer = 2  
Const PosFile As Integer = 3  
Dim FileNumber As Integer  
Dim Posj As Integer  
Dim TempStr As String  
Dim TempTimeLng As Long
```

```
Posj = 2
```

```
‘открываем файл на чтение  
FileNumber = FreeFile()  
Open FileName For Input As FileNumber
```

```
‘просматриваем весь файл до конца  
Do While Not EOF(FileNumber)
```

```
‘считываем построчно
```

```
Line Input #FileNumber, TempStr
```

```
If Len(TempStr) > 0 Then
```

```
With Sheets(«Playback-list»)
```

```
‘вызываем функцию проверки: является ли считанная запись
```

```
‘записью о фильме
```

```
If FindMovie(TempStr) = True Then
```

```
.Cells(Posj, PosDate) = GetDate(TempStr)
```

```
.Cells(Posj, PosTime) = GetTimeLng(GetTime(TempStr))
```

```
.Cells(Posj, PosFile) = GetFile(TempStr)
```

```
Posj = Posj + 1
```

```
End If
```

```
End With
```

```
End If
```

```
Loop
```

```
Close FileNumber
```

```
End Function
```

```
‘функция проверяет: является ли переданная строка записью о
```



фильме,
'возвращает "TRUE", если запись о фильме
Function FindMovie(str As String) As Boolean

```
Const MarkMovie As String = "type=""Movie""  
Const MarkMovieWav As String = "type=""Movie_Wav""  
Dim exst As Integer
```

```
exst = InStr(1, str, MarkMovie, vbTextCompare)  
If exst = 0 Then  
    exst = InStr(1, str, MarkMovieWav, vbTextCompare)  
    If exst = 0 Then  
        FindMovie = False  
        Exit Function  
    End If  
End If  
FindMovie = True  
End Function
```

'функция считывает из строки "SrcStr" данные
'StrBeg – параметр, указывающий,
'после какой подстроки нужно считывать текст
'StrEnd – параметр, указывающий,
'до какой подстроки нужно считывать текст
'Возвращает считанную подстроку
Function GetData(SrcStr As String, StrBeg As String, StrEnd As String) As
String

```
Dim PosBegin As Integer  
Dim PosEnd As Integer  
Dim LenMarkData As Integer
```

```
PosBegin = InStr(1, SrcStr, StrBeg, vbTextCompare)  
If PosBegin = 0 Then  
    GetData = ""  
    MsgBox "Data is not present"  
    Exit Function  
Else  
    LenMarkData = Len(StrBeg)  
    PosBegin = PosBegin + LenMarkData  
End If  
PosEnd = InStr(PosBegin, SrcStr, StrEnd, vbTextCompare)  
If PosEnd = 0 Then  
    GetData = ""  
    MsgBox "Data is not present"
```



```
Exit Function
Else
    GetData = Mid(SrcStr, PosBegin, PosEnd - PosBegin)
End If
End Function
```

```
'функция, устанавливающая параметр,
'указывающий, после какой подстроки
'нужно считывать текст – название файла,
' и параметр, указывающий, до какой подстроки
'нужно считывать текст,
'затем вызывает функцию, возвращающую название файла
Function GetFile(str As String) As String
    Dim MarkFileBegin As String
    Dim MarkFileEnd As String
    MarkFileBegin = " file="
    MarkFileEnd = ""
    GetFile = GetData(str, MarkFileBegin, MarkFileEnd)
End Function
```

```
'функция, устанавливающая параметр, указывающий,
'после какой подстроки
'нужно считывать текст – дату старта,
'и параметр, указывающий,
'до какой подстроки нужно считывать текст,
'затем вызывает функцию, возвращающую название файла
Function GetDate(str As String) As String
    Dim MarkFileBegin As String
    Dim MarkFileEnd As String
    MarkFileBegin = " date="
    MarkFileEnd = ""
    GetDate = GetData(str, MarkFileBegin, MarkFileEnd)
End Function
```

```
'функция, устанавливающая параметр, указывающий,
'после какой подстроки нужно
'считывать текст – время старта,
'и параметр, указывающий,
'до какой подстроки нужно считывать текст
'затем вызывает функцию, возвращающую название файла
Function GetTime(str As String) As String
    Dim MarkFileBegin As String
    Dim MarkFileEnd As String
    MarkFileBegin = " time="
    MarkFileEnd = ""
```



```
    GetTime = GetData(str, MarkFileBegin, MarkFileEnd)
End Function
```

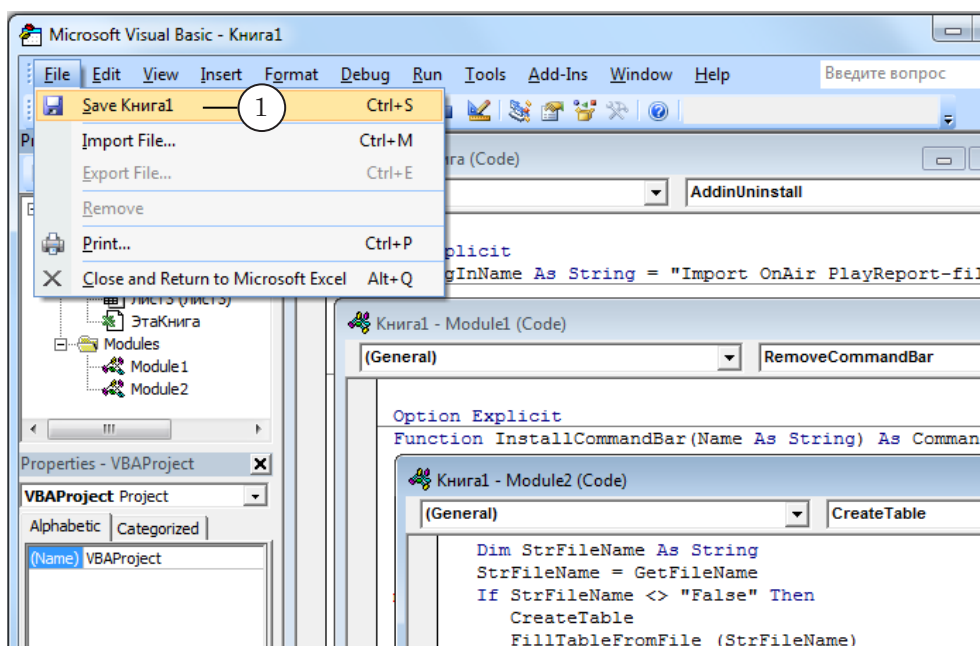
```
'функция, преобразует строковое значение времени в целое число
Public Function GetTimeLng(str As String) As Long
```

```
On Error GoTo Errorlabel
Const LenTime As Integer = 11
Dim StrTime As String
Dim ttime, ResTime, z As Long
Dim tstr, sep As String
Dim posb, pose As Integer

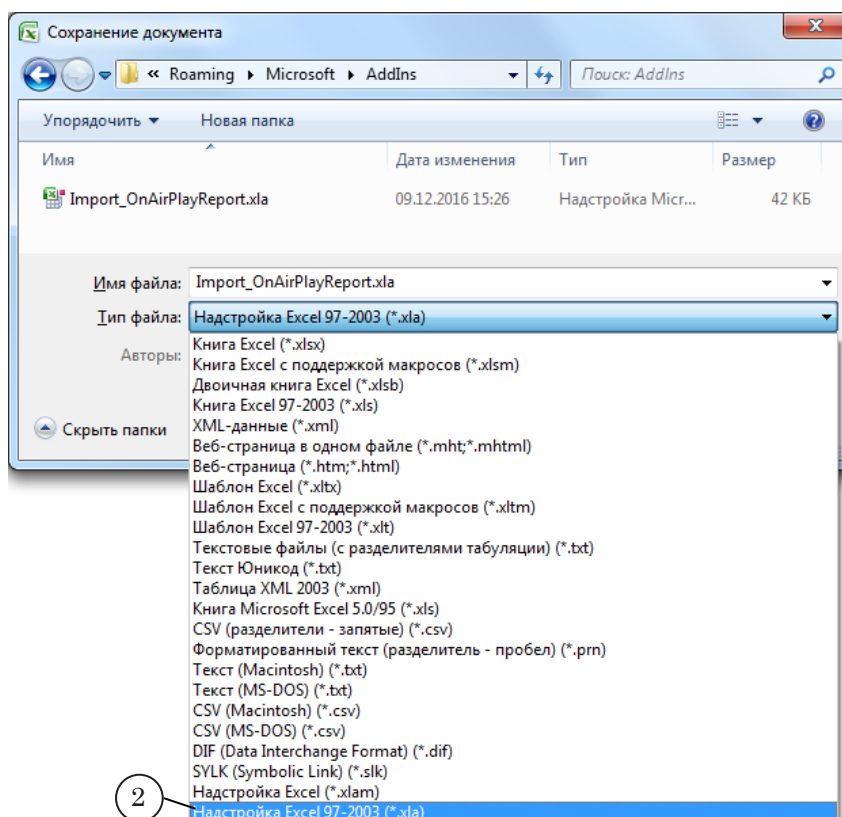
StrTime = Left(str, LenTime)
posb = 1
pose = 1
sep = ":"
z = 1000000
StrTime = StrTime & "."
ttime = 0
pose = InStr(posb, StrTime, sep, vbTextCompare)
Do While pose <> 0
    tstr = Mid(StrTime, posb, pose - posb)
    posb = pose + 1
    ttime = ttime + CLng(tstr) * z
    z = z / 100
    pose = InStr(posb, StrTime, sep, vbTextCompare)
    If pose = 0 Then
        pose = posb
        sep = "."
        pose = InStr(posb, StrTime, sep, vbTextCompare)
    End If
Loop
GetTimeLng = ttime
Exit Function
Errorlabel:
    GetTimeLng = -1
End Function
```



6. Сохраните проект макроса в виде отдельного файла. Для этого в меню редактора Microsoft Visual Basic выберите команду File > Save Книга1 (1).



Откроется диалог записи файлов Сохранение документа, в котором выберите тип записываемого файла Надстройка Excel с расширением *.xla (2), и сохраните его, присвоив ему имя. В рассматриваемом примере файл с макросом имеет название Import_OnAirPlayReport.xla.

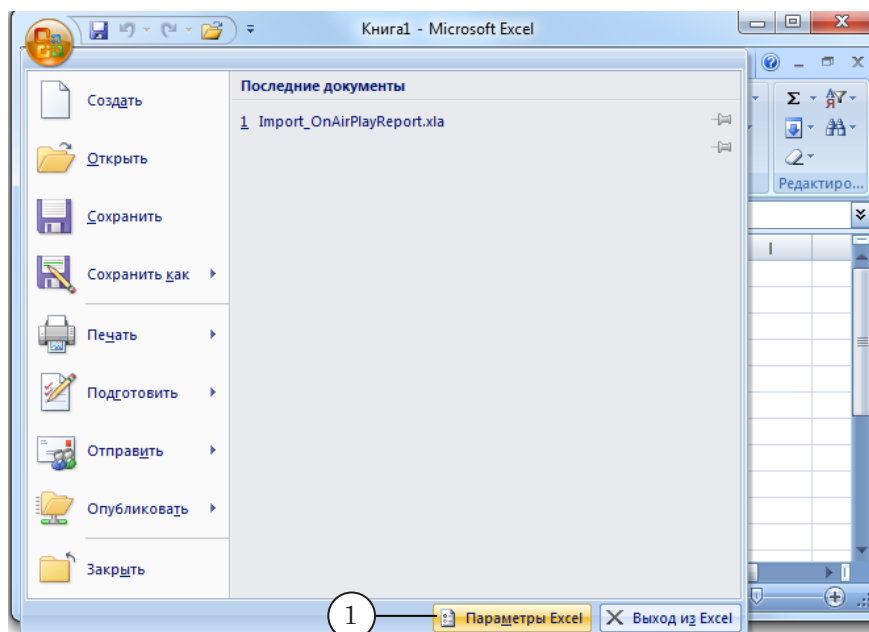




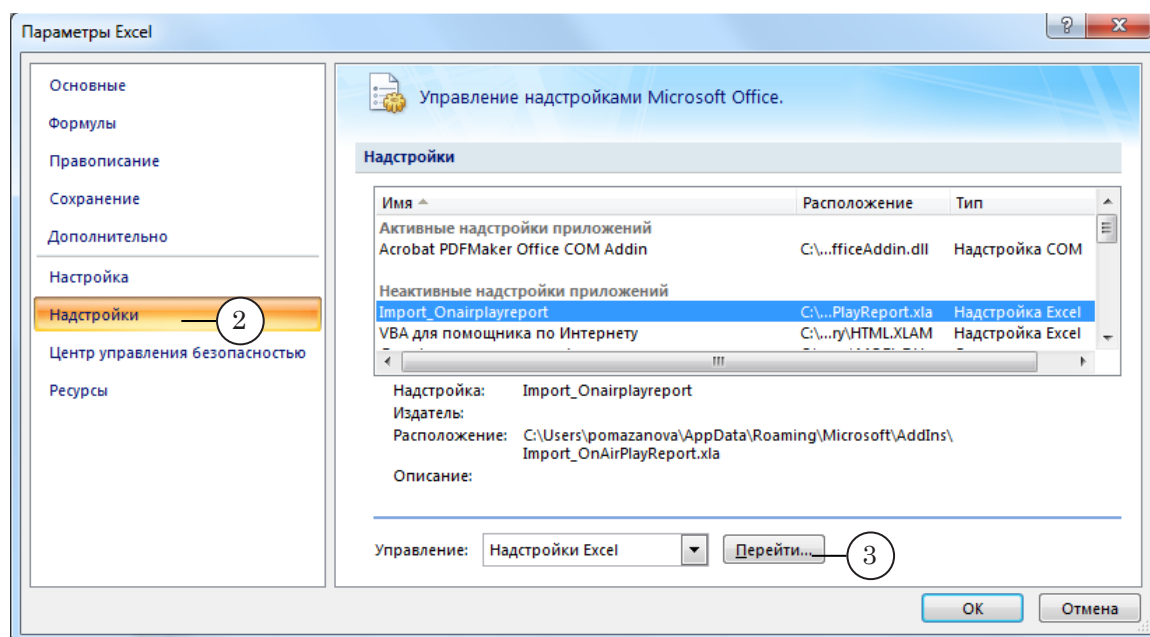
Подключение и использование макроса в приложении Microsoft Excel

Чтобы подключить созданный макрос из приложения Microsoft Excel и работать с ним для составления отчёта о воспроизведённых в эфир фильмах, выполните следующие шаги:

1. В главном меню приложения Microsoft Excel нажмите кнопку Параметры Excel (1).

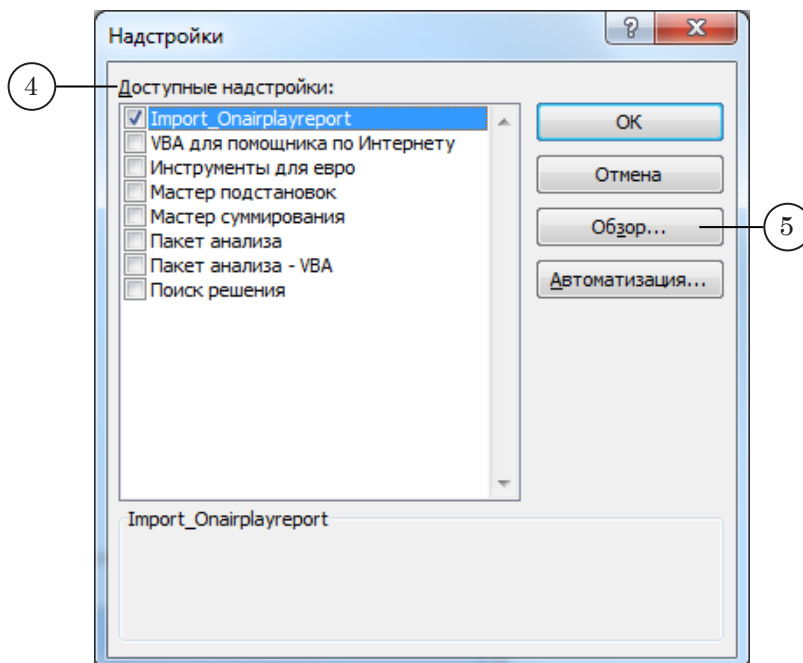


2. Вызовите диалог Надстройки (2), из которого и производится подключение программ-макросов. В окне Управление надстройками Microsoft Office нажмите кнопку Перейти (3).





3. Если в списке Доступные надстройки (4) отсутствует необходимый макрос с названием `Import_Onairplayreport`, внесите его в список, нажав на кнопку Обзор (5) и указав в появившемся окне диалога путь к файлу `Import_OnAirPlayReport.xla`.



4. Нажмите ОК. После включения макроса на панели инструментов приложения Microsoft Excel появится новая кнопка `Import OnAir PlayReport-file`.
5. Нажатие на эту кнопку вызывает диалог выбора файлов в формате XML с расширением `*.PlayReport`. В этом диалоге необходимо выбрать имя файла `airMMДД.PlayReport` со списком команд, выполненных программой `FDOOnAir` за отчётный период. После загрузки файла текущий лист таблицы макрос автоматически переименует в `Playbacklist`, который будет содержать информацию о воспроизведённых в эфир фильмах:
 - первая колонка `Date` – дата воспроизведения;
 - вторая `Time` – время начала воспроизведения;
 - третья `File` – путь к файлу и его название.

Таким образом, результатом работы макроса в данном случае будет являться файл-таблица в формате Microsoft Excel со списком воспроизведённых в `FDOOnAir` видеофайлов, содержащая дату и время начала воспроизведения.



Полезные ссылки

Линейка продуктов ФорвардТ: описание, загрузка ПО, документация, готовые решения

<http://www.softlab-nsk.com/rus/forward>

Техподдержка

e-mail: forward@softlab.tv

forward@sl.iae.nsk.su

forward@softlab-nsk.com

Форумы

<http://www.softlab-nsk.com/forum>

Документы, рекомендованные в данном руководстве для дополнительного ознакомления:

1. [FDOнAir. Автоматизация вещания;](#)
2. [Лог-файлы FDOнAir. Настройка записи и просмотр.](#)