

***Создание отчёта  
о воспроизведённых в  
FDOпAir  
видеоданных  
с помощью макроса  
в приложении  
Microsoft Excel***

Версия от 07 апреля 2005

ForwardT Software 3.7.0

# Содержание

<b>СОДЕРЖАНИЕ .....</b>	<b>2</b>
<b>1 ВВЕДЕНИЕ .....</b>	<b>3</b>
<b>2 НАСТРОЙКА ПРОГРАММЫ <i>FDONAIR</i> .....</b>	<b>4</b>
<b>3 СОЗДАНИЕ МАКРОСА НА ЯЗЫКЕ VISUAL BASIC В ПРИЛОЖЕНИИ <i>MICROSOFT EXCEL</i>.....</b>	<b>5</b>
<b>4 ПОДКЛЮЧЕНИЕ И ИСПОЛЬЗОВАНИЕ МАКРОСА В ПРИЛОЖЕНИИ <i>MICROSOFT EXCEL</i>.....</b>	<b>12</b>

# 1 Введение

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

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

## 2 Настройка программы *FDO*nAir

Для составления отчёта о телевещании с помощью программы *FDO*nAir необходим список выполненных заданий. Такой список создается автоматически в процессе работы программы, если перед началом вещания включить опцию **Писать список выполненных команд (Enable playback reports)** на панели **Журнал (Log-book)** диалога настроек *Установки (Settings)*. Этот диалог открывается из основного окна приложения кнопкой **Установки (Settings)**. Более подробно это описано в руководстве пользователя приложением *FDO*nAir.

На этой же панели с помощью кнопки **Каталог списка (Reports folder)** 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*nAir 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 разделе 7.1 руководства пользователя приложением *FDO*nAir.

### 3 Создание макроса на языке Visual Basic в Приложении *Microsoft Excel*

Чтобы составить отчёт о телевещании за определённый интервал времени с помощью списка выполненных заданий из отчёта **airММДД.PlayReport** в программе *FDonAir*, нужно выбрать файлы заданного типа. Например, это могут быть только фильмы – файлы с расширением \*.avi; \*.tml; \*.mpg; \*.mpeg; \*.mpv; \*.m2v; \*.m2p; \*.mp2, или только статичные изображения – файлы с расширением \*.tga;\*.bmp;\*.dib;\*.jpg;\*.png;\*.422. В данном разделе описано, как можно создать макрос на языке **Visual Basic** из приложения *Microsoft Excel*, который позволит автоматизировать процедуру выбора только фильмов из списка воспроизведённых в *FDonAir* заданий.

Для создания макроса нужно из меню **Tools** приложения *Microsoft Excel* вызвать диалог *Macro*, в котором в поле **Macro name:** задать имя макроса, а затем выполнить команду **Create**. Такая последовательность действий приводит к открытию окна редактора *Microsoft Visual Basic*, в который нужно скопировать тексты функций и процедур на языке **Visual Basic**, приведённые ниже.

Двойной щелчок левой кнопкой мыши по одному из подразделов схемы проекта созданного макроса **VBAProject**, расположенных в верхнем левом углу окна редактора, открывает справа поле этого подраздела. В это поле можно вносить текст программы и затем редактировать его необходимым образом. Проект создаваемого макроса **VBAProject** по умолчанию включает в себя 2 раздела: **Microsoft Excel Objects** с подразделами **Sheet1 (Sheet1) (Лист 1 (Лист 1))**, **Sheet2 (Sheet2) (Лист 2 (Лист 2))**, **Sheet3 (Sheet3) (Лист 3 (Лист 3))**, **ThisWorkbook (Эта книга)** и раздел **Modules** с одним подразделом **Module1**. Если раздел **Modules** или включённый в него подраздел **Module1** отсутствуют, их можно добавить командой **Module** из меню **Insert** редактора.

Текст макроса для автоматизации выбора фильмов из списка воспроизведённых в *FDonAir* файлов будет состоять из трёх частей. Первая 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 подраздела **ThisWorkbook (Эта книга)** раздела **Microsoft Excel Objects**, два раза щёлкнув по нему левой кнопкой мыши.

Ниже приведена часть макроса, которая отвечает за инициализацию кнопки **Import OnAir PlayReport-file** при включении макроса и за удаление её после его отключения:

```
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
```

Две другие части макроса – исполнимые модули, которые в данном примере будут называться **ScanOnAirPlayReportCommandBar** и **ScanOnAirPlayReportXml** и являться подразделами раздела **Modules**. Они создаются предварительно с помощью команды **Insert** командного или контекстного меню. Затем в их поля копируются соответствующие тексты процедур.

Модуль **ScanOnAirPlayReportCommandBar** (текст ниже) отвечает за создание кнопки **Import OnAir PlayReport-file**:

```

Option Explicit
Function InstallCommandBar(Name As String) As CommandBar

    Dim NewBtn As CommandBarButton

    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
ErrHandler:
    MsgBox Err.Description

End Sub

```

Функции и процедуры модуля **ScanOnAirPlayReportXml** осуществляют работу с выбранным файлом **airММДД.PlayReport**:

- ` 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

```

```

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
```

После копирования частей текста макроса в соответствующие разделы проект макроса нужно сохранить в виде отдельного файла. Для этого из командного меню редактора **File** нужно вызвать диалог записи файлов *Save Book*, в котором выбрать тип записываемого файла **Save as type** как **Microsoft Excel Add-In (\*.xla)**, и сохранить его, присвоив ему имя. В рассматриваемом примере файл с макросом имеет название **Import\_OnAirPlayReport.xla**.

## 4 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, нужно из меню **Tools** приложения *Microsoft Excel* вызвать диалог *Add-Ins*, из которого и производится подключение программ-макросов. Если в списке **Add-Ins Available:** отсутствует необходимый макрос с названием **Import\_Onairplayreport**, то нужно внести его в список, нажав на кнопку **Browse...** и указав в появившемся окне диалога путь к файлу **Import\_OnAirPlayReport.xla**.

После включения макроса на панели инструментов приложения *Microsoft Excel* должна появиться новая кнопка **Import OnAir PlayReport-file**. Нажатие на эту кнопку вызывает диалог выбора файлов в формате XML с расширением **\*.PlayReport**. В этом диалоге необходимо выбрать имя файла **airММДД.PlayReport** со списком команд, выполненных программой *FDO nAir* за отчётный период. После загрузки файла текущий лист таблицы макрос автоматически переименует в **Playback-list**, который будет содержать информацию о воспроизведённых в эфир фильмах: первая колонка **Date** – дата воспроизведения, вторая **Time** – время начала воспроизведения, а 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 *FDO nAir* видео файлов, содержащая дату и время начала воспроизведения.