

***Создание файла со списком  
заданий для титровального  
элемента «Ролики»  
с помощью макроса в  
приложении Microsoft Excel***

Версия от 01 февраля 2005  
ForwardT Software 3.6.0

# Содержание

СОДЕРЖАНИЕ .....	2
1 ВВЕДЕНИЕ .....	3
2 СОЗДАНИЕ МАКРОСА НА ЯЗЫКЕ <i>VISUAL BASIC</i> В ПРИЛОЖЕНИИ <i>MICROSOFT EXCEL</i> ДЛЯ ПРЕОБРАЗОВАНИЯ СПИСКА ЗАДАНИЙ ТИТРОВАЛЬНОГО ЭЛЕМЕНТА <i>РОЛИКИ</i> ИЗ ТАБЛИЦЫ <i>MICROSOFT EXCEL</i> В ФАЙЛ ФОРМАТА XML.....	4
3 ПОДКЛЮЧЕНИЕ И ИСПОЛЬЗОВАНИЕ МАКРОСА В ПРИЛОЖЕНИИ <i>MICROSOFT EXCEL</i> .....	9
4 СОСТАВЛЕНИЕ СПИСКА ЗАДАНИЙ ДЛЯ ТИТРОВАЛЬНОГО ЭЛЕМЕНТА <i>РОЛИКИ</i> В ЭЛЕКТРОННОЙ ТАБЛИЦЕ <i>MICROSOFT EXCEL</i> .....	10

# 1 Введение

В данном документе предложен альтернативный способ создания файла списка заданий для титровального элемента *MovieList (Полики)*, который может быть включён в титровальный проект, созданный в приложении *FDTitleDesigner* и воспроизведён в эфире. Файл может быть сформирован на основе списка, составленного в электронной таблице *Microsoft Excel* с помощью программы-макроса, написанной на языке **Visual Basic** для *Приложений (VBA)*. Ниже приведён текст макроса, описана последовательность действий, которые необходимо произвести для создания и подключения макроса в приложении *Microsoft Excel*, а также изложены правила составления списка заданий для титровального элемента *Полики* в таблице *Microsoft Excel* и рассмотрен пример использования макроса – преобразование списка заданий из таблицы *Microsoft Excel* в файл формата XML.

## 2 Создание макроса на языке *Visual Basic* в приложении *Microsoft Excel* для преобразования списка заданий титровального элемента *Ролики* из таблицы *Microsoft Excel* в файл формата XML

Для создания программы-макроса на языке **Visual Basic** из приложения *Microsoft Excel*, которая позволяет преобразовывать список заданий для титровального элемента *Ролики* из таблицы *Microsoft Excel* в файл формата XML, необходимо из меню **Tools** приложения *Microsoft Excel* вызвать диалог **Macro**. В этом диалоге в поле **Macro name:** требуется задать имя макроса, а затем выполнить команду **Create**. В результате откроется окно *Microsoft Visual Basic*, в которое нужно скопировать тексты функции 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)**.
- 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, позволяющего создавать XML-файл со списком заданий для титровального элемента *Ролики* на основе списка заданий в таблице *Microsoft Excel*, будет состоять из двух частей. Первая часть включает функцию, создающую при включении макроса кнопку **Create SLMovieList-file** на панели рабочей области окна приложения *Microsoft Excel* и процедуру, удаляющую эту кнопку при отключении макроса. Созданная кнопка позволит вызвать диалог для сохранения сформированного файла со списком заданий для титровального элемента *Ролики* в формате XML с расширением \*.SLMovieList. Поскольку кнопка **Create SLMovieList-file** должна быть выведена именно в рабочей области, эта часть макроса должна быть связана непосредственно с рабочей областью приложения. Поэтому текст функции, приведённый ниже, следует скопировать в поле подраздела **ThisWorkbook (Эта книга)** раздела **Microsoft Excel Objects**, два раза щёлкнув по нему левой кнопкой мыши.

```
Option Explicit
Const PlugInName As String = "Create SLMovieList-file"

Private Sub Workbook_AddinInstall()
    InstallCommandBar (PlugInName)
End Sub

Private Sub Workbook_AddinUninstall()
    RemoveCommandBar (PlugInName)
End Sub

Function InstallCommandBar(Name As String) As CommandBar

    Dim NewBtn As CommandBarControl
    ' Проверка на наличие ошибок, например,
    'если кнопка с таким именем уже существует
    On Error GoTo ErrHandler
```

```

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

' Добавляется кнопка в рабочей области приложения
    Set NewBtn = InstallCommandBar.Controls.Add(msoControlButton)
    NewBtn.Style = msoButtonCaption
    NewBtn.Caption = " Create SLMovieList-file"
    NewBtn.OnAction = "Module1.WriteToXML_Sample"

'Показывается панель инструментов
    InstallCommandBar.Visible = True

Exit Function
ErrorHandler:

    MsgBox Err.Description

End Function

Sub RemoveCommandBar(Name As String)
    On Error GoTo ErrorHandler
    Application.CommandBars(Name).Delete
    Exit Sub

ErrorHandler:
    MsgBox Err.Description

End Sub

```

Второй частью макроса является процедура, которая преобразует список заданий из формата таблицы *Microsoft Excel* в файл с расширением \*.SLMovieList. Ниже приведён текст процедуры с поясняющими комментариями, который необходимо скопировать в поле подраздела **Module1** в разделе **Modules**. Название модуля **Module1** менять не следует, поскольку модуль именно с таким именем вызывается для исполнения по нажатию созданной кнопки **Create SLMovieList-file**.

```

Sub WriteToXML_Sample()
'
'
' 1) Вызываем файловый диалог для выбора имени файла
'
Dim fileSaveName As String
fileSaveName = Application.GetSaveAsFilename( _
fileFilter:="Movie List Files (*.SLMovieList),
*.SLMovieList", _
Title:="Save", _
InitialFileName:="" _
)

```

```

'если в диалоге отказаться от выбора, процедура завершится
If fileSaveName = "False" Then
    Exit Sub
End If
'
'
'2) Открываем новый XML-документ с указанным именем
' с помощью спец-объекта (MXXMLWriter)
'
'Создаём объект MXXMLWriter.
Dim Writer As Object
Set Writer = CreateObject("MSXML2.MXXMLWriter")

' Устанавливаем форматирование документа для читаемости XML.
Writer.indent = True

'Ссылаемся на документ через интерфейс IVBSAXContentHandler,
'получаем доступ к содержимому
'(Приведение к интерфейсу IVBSAXContentHandler
'осуществляется автоматически)
Dim Content As Object
Set Content = Writer
'Content - содержимое, с ним дальше и будем работать
'
'
' 3) Формирование документа
' на каждую строку:
' проверяем, что в первой колонке корректный тип,
' открываем новую запись в XML-файле с этим типом
' и параметром CellDuration из третьей колонки
' (если CellDuration не равно 0)
' и добавляем в тело записи (value) значение из второй колонки:
' для movie и для image - имя файла, для pause - ничего
'
Content.startDocument 'Указываем начало документа
'Создаём объект атрибутов
Dim Attributes As Object
Set Attributes = CreateObject("MSXML2.SAXAttributes")

'Заполняем атрибуты для корневого элемента movieList
Attributes.addAttribute "", "", "xmlns", "", "http://x-
schema:movieListSchema"
Attributes.addAttribute "", "", "TCF", "", "sec"

'Создаём корневой элемент <movieList>
Content.startElement "", "", "movieList", Attributes
' что должно получиться -

```

```

'(<movieList xmlns="http://x-schema:movieListSchema" TCF="sec">)

'На каждую ячейку первого столбца текущего листа:
Dim cell As Variant
For Each cell In ActiveSheet.Columns(1).Cells

    'берём имя элемента (CellElementName) из первой колонки
    Dim CellElementName As String
    CellElementName = cell.text

    Select Case CellElementName
        'если оно совпадает с одним из следующих, начинаем обработку:
        Case "movie", "image", "pause"

            'чистим атрибуты, чтобы не остались предыдущие
            Attributes.Clear

            'берём длительность (CellDuration) из третьей колонки
            Dim CellDuration As String
            CellDuration = cell.Next.Next.text

            ' если Длительность задана и это не "movie"
            ' (длительность ролика задать нельзя)
            If CellDuration <> "" And cell.text <> "movie" Then

                ' Добавляем её в атрибуты (Attributes)
                Attributes.addAttribute "", "", "duration", "",
CellDuration

            End If

            'Открываем элемент с соответствующим
            'именем <CellElementName>
            Content.startElement "", "", CellElementName,
Attributes

            ' если элемент не Пауза - <pause>
            If cell.text <> "pause" Then

                'открываем элемент <value>
                Content.startElement "", "", "value", Nothing

                ' берём источник (CellSourceName) из второй колонки
                Dim CellSourceName As String
                CellSourceName = cell.Next.text
                'и прописываем его в тело элемента Value
                Content.Characters CellSourceName

```

```

        'Закрываем элемент </value>
        Content.endElement "", "", "value"

    End If

    'Закрываем элемент с соответствующим
    'именем </CellElementName>
    Content.endElement "", "", CellElementName

End Select
Next

' Закрываем корневой элемент </movieList>
Content.endElement "", "", "movieList"

'Закрываем документ
Content.endDocument
'
'-----
' 4) Созданный в памяти XML-документ записываем в файл,
'      выбранный в начале процедуры.
'-----
' Создаём объект управления файлами
Dim FileSystem As Object
Set FileSystem = CreateObject("Scripting.FileSystemObject")

'Открываем текстовый файл с указанным ранее именем "fileSaveName"
Dim TextStream As Object
Set TextStream =
FileSystem.CreateTextFile(Filename:=fileSaveName, Overwrite:=True,
Unicode:=True)

'Пишем в файл содержимое XML документа
TextStream.Write (Writer.output)

'закрываем файл
TextStream.Close

End Sub
'-----

```

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 определить тип записываемого файла (**Save as type**) как **Microsoft Excel Add-In (\*.xla)**, присвоить макросу имя и сохранить.



### 3 Подключение и использование макроса в приложении *Microsoft Excel*

Чтобы подключить созданный макрос из приложения *Microsoft Excel* и работать с ним для преобразования списка заданий, составленного в таблице *Microsoft Excel*, в файл формата XML, 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, то внести его в список можно, нажав на кнопку **Browse...** и указав в появившемся окне диалога путь к требуемому файлу-макросу с расширением **\*.xla**.

После включения макроса на панели инструментов приложения *Microsoft Excel* должна появиться новая кнопка **Create SLMovieList-file**. Нажатие на эту кнопку вызывает диалог для сохранения файлов в формате XML с расширением **\*.SLMovieList**. В этом диалоге при дальнейшей работе по преобразованию списка заданий в формат XML будет указываться имя создаваемого XML-файла.

## 4 Составление списка заданий для титровального элемента *Ролики* в электронной таблице *Microsoft Excel*


Список заданий для титровального элемента *Ролики*, который в дальнейшем будет преобразован в файл формата XML, нужно составить предварительно в таблице *Microsoft Excel*. Поскольку преобразование списков заданий в файлы будет выполняться с помощью программы-макроса, необходимо эти списки составлять в соответствии с определёнными правилами, о которых и будет рассказано в этом разделе. Список-таблица должен содержать три колонки ячеек формата **General**: в первой (**A**) указан тип задания (**movie**, **image** или **pause**), во второй (**B**) – имя файла (для **movie** или **image**) и в третьей (**C**) – длительность задания (в секундах).

- В первой колонке (**A**) допустимо указывать один из трёх типов задания:

**movie** – фильм;

**image** – статичное изображение;

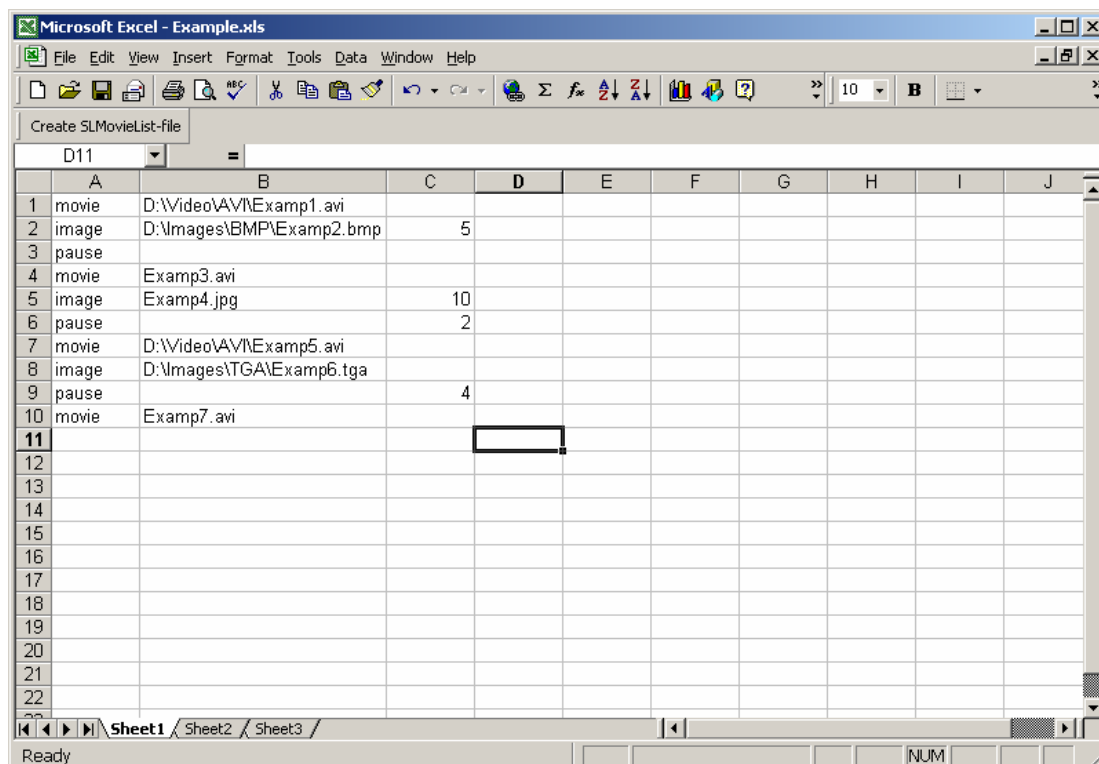
**pause** – пауза.

 При указании типа задания с заглавной буквы макрос не сможет опознать и обработать строку задания!

- Во второй колонке (**B**) должны быть указаны пути к имени файла задания (относительные или абсолютные). Путь для задания типа **movie** – это путь к AVI-файлу, для задания типа **image** – путь к файлу со статичным изображением (\*.tga, \*.png, \*.bmp, \*.dib, \*.jpg). Для задания **pause** файл создавать не нужно и, соответственно, не нужно вносить значение в ячейку этой колонки.
- Третья колонка (**C**) должна содержать значения длительности заданий (в секундах). Заданием **movie** является AVI-файл, длительность которого определена изначально, поэтому значение ячейки в этой колонке для заданий типа **movie** определять не нужно.

Подробный формат файла \*.SLMovieList и описание титровального элемента **MovieList (Ролики)** можно найти в руководстве пользователя программой *FDTitleDesigner*.

На Рис. 1 приведён пример списка заданий для титровального элемента *Ролики*, составленного в соответствии с описанными выше правилами.



	A	B	C	D	E	F	G	H	I	J
1	movie	D:\Video\AVI\Examp1.avi								
2	image	D:\Images\BMP\Examp2.bmp	5							
3	pause									
4	movie	Examp3.avi								
5	image	Examp4.jpg	10							
6	pause		2							
7	movie	D:\Video\AVI\Examp5.avi								
8	image	D:\Images\TGA\Examp6.tga								
9	pause		4							
10	movie	Examp7.avi								
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										

Рис. 1. Пример списка заданий для титровального элемента *Ролики* в таблице *Microsoft Excel*

Если после преобразования макросом списка заданий, приведённого на Рис. 1, созданный файл в формате XML с расширением \*.SLMovieList открыть при помощи какого-нибудь текстового редактора, например, *Notepad.exe*, то в нём будут содержаться следующие записи:

```
?xml version="1.0" encoding="UTF-16" standalone="no"?>
<movieList xmlns="http://x-schema:movieListSchema" TCF="sec">
  <movie>
    <value>D:\Video\AVI\Examp1.avi</value>
  </movie>
  <image duration="5">
    <value>D:\Images\BMP\Examp2.bmp</value>
  </image>
  <pause/>
  <movie>
    <value>Examp3.avi</value>
  </movie>
  <image duration="10">
    <value>Examp4.jpg</value>
  </image>
  <pause duration="2"/>
  <movie>
    <value>D:\Video\AVI\Examp5.avi</value>
  </movie>
  <image>
    <value>D:\Images\TGA\Examp6.tga</value>
  </image>
  <pause duration="4"/>
  <movie>
    <value>Examp7.avi</value>
  </movie>
</movieList>
```

Файл, созданный таким образом, можно открывать и просматривать с помощью специального редактора заданий *FDMovieListEditor* для титровального элемента **MovieList (Полики)** или включать в титровальный проект, создаваемый в приложении *FDTitleDesigner*.