Report creation about data reproduced in FDOnAir by macro in Microsoft Excel

User's Guide

Revision from 11 April 2005
ForwardT Software 3.7.0

SoftLab-NSK, Ltd.

Notice

The information in this document is subject to change without prior notice in order to improve reliability, design, or function and does not represent a commitment on the part of this company.

In no event will we be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or the inability to use the product or documentation, even if advised of the possibility of such damages.

Throughout this manual, we make reference to product names that are trademarks of other companies. We are using these names for identification purposes only, with no intention of infringement of the trademarks.

FCC Information

FCC ID:

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Shielded cables and I/O cards must be used for this equipment to comply with the relevant FCC regulations. Changes or modifications not expressly approved in writing by SoftLab-NSK, Ltd. may void the user's authority to operate this equipment.

Limited Warranty

Our company warrants this product against defects in materials and workmanship for a period of one year from the date of purchase. During the warranty period, products determined by us to be defective in form or function will be repaired or replaced at our option, at no charge. This warranty does not apply if the product has been damaged by accident, abuse, misuse, or as a result of service or modification other than by us.

This warranty is in lieu of any other warranty expressed or implied. In no event shall we be held liable for incidental or consequential damages, such as lost revenue or lost business opportunities arising from the purchase of this product.

Table of Contents

TABLE OF CONTENTS 3

1 INTRODUCTION 4

2 FDONAIR SETTINGS 5

3 MACRO CREATION BY *VISUAL BASIC* IN *MICROSOFT EXCEL* 6

4 USE OF MACRO IN *MICROSOFT EXCEL* 13

1 Introduction

In this document the example of report creation about play backed data for time period with help **Log-book** of *FDOnAir* is described. For a basis the file created in *FDOnAir* and macro are used. The file contains the commands list executed by the program during work. Macro is written on programming language **Visual Basic** for *Application (VBA)*.

The macro text is below given. The sequence of actions on the appropriate adjustments of work *FDOnAir*, creation and connection of macro from *Microsoft Excel* (*Microsoft Office*) and the subsequent work with it are described.

2 FDOnAir Settings

The report on broadcasting by *FDOnAir* playback report list with information on executed commands is necessary. Such list is created in the dialog *Program log-book/Playback report* if the option **Enable playback reports** in the panel **Log-book** of *Settings* dialog is switched on. In more detail it is described in the *FDOnAir* User's Guide.

In same panel it is necessary to select a folder for file record by the button **Reports folder**. The file name will be defined as **airMMDD.PlayReport** according to the current date – month (MM) and day (DD) of the program work. The file is organized in a format XML. At work of *FDOnAir* to a file records about names of play backed files, date and time of the beginning and other information will be saved. More detailed information about structure of file **airMMDD.PlayReport** and types of the commands, which are written down it, it is possible to find in section 7.1 of the *FDOnAir* User's Guide.

3 Macro creation by *Visual Basic* in *Microsoft Excel*

To make report about broadcasting for certain time period in *FDOnAir* by it is possible to select from file **airMMDD.PlayReport** files with fixed types, for example, only movies – files with extension *.avi; *.tml; *.mpg; *.mpeg; *.mpv; *.m2v; *.m2p; *.mp2, or only pictures – files with extension *.tga;*.bmp;*.dib;*.jpg;*.png;*.422. In this section it is described, as it is possible to create macro in language **Visual Basic** from application *Microsoft Excel*. This macro allows automating procedure of a choice from the broadcasting report only files with movies.

For macro creation it is necessary to open dialog *Macro* from menu **Tools** of *Microsoft Excel* application. In the field **Macro name:** it is necessary to set a macro name and then to execute command **Create**. Such sequence of actions results in opening of *Microsoft Visual Basic* editor window. Described below the texts of functions and procedures in language **Visual Basic** need to be copied in this dialog.

Double click by the mouse left button on one of subsection of the project scheme of created macro **VBAProject**, located in the top left corner of editor window, opens on the right a field of this subsection. The program text is inserted to this field and then it is edited.

The project of created macro **VBAProject** includes on default 2 sections: **Microsoft Excel Objects** with subsections **Sheet1 (Sheet1)**, **Sheet2 (Sheet2)**, **Sheet3 (Sheet3)**, **ThisWorkbook** and section **Modules** with subsection **Module1**. If section **Modules** or subsection **Module1** are absent, they can be added by the command **Module** from menu **Insert** of the editor.

The macro text for automation of movies select from playback report of *FDOnAir* contains three parts. The first part includes commands for creation of button **Import OnAir PlayReport-file** on the panel of working area of application *Microsoft Excel* window. This button allows to open dialog of select of tasks file executed by program *FDOnAir* **airMMDD.PlayReport**. As button **Import OnAir PlayReport-file** should be displayed on the panel of working area this macro part should be connected directly to working area of the application. Therefore the text given below, it is necessary to copy to a field of subsection **ThisWorkbook** of the section **Microsoft Excel Objects**, two times having clicked on it by the mouse left button.

The part of macro, which is responsible for initialization of the button **Import OnAir PlayReport-file** at macro start and for it's deleting at macro stop is below given:

```
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
```

The two other macro parts – executed modules, which in this example will be named **ScanOnAirPlayReportCommandBar** and **ScanOnAirPlayReportXml**. They will be subsections of section **Modules**. They are created preliminary by the command **Insert** of the command or contextual menu. Then the appropriate texts of procedures are copied to their fields.

Module **ScanOnAirPlayReportCommandBar** (text below) is responsible for creation of the button **Import OnAir PlayReport-file**:

```
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

ErrHandler:
    MsgBox Err.Description

End Sub
```

The functions and procedures of module **ScanOnAirPlayReportXml** make with selected file **airMMDD.PlayReport**:

```
‘  Control procedure, starts consecutively procedures for file opening
‘  and table completion
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

‘  Function of open of select file dialog
‘  restore name of selected file
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

‘  Function for table creation, sets name of sheet and columns,
‘  format of column «Time»,
‘  in which it is possible to write down only an integer
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

‘  Function which opens a file, reads out from it data
‘  and fills in columns of the table
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
   ‘open the file for reading
    FileNumber = FreeFile()
```

```
    Open FileName For Input As FileNumber

   ‘view all file up to the end
    Do While Not EOF(FileNumber)
   ‘read out file on line
       Line Input #FileNumber, TempStr
       If Len(TempStr) > 0 Then
          With Sheets("Playback-list")
          ‘cause check function: whether the read out record –
          ‘record about movie
              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

‘function checks up whether transmitted string record about movie,
‘return «TRUE», if movie
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

‘function reads out data from string «SrcStr»
‘StrBeg – parameter indicating,
‘after what substring it is necessary to read out text
```

```
‘StrEnd – parameter indicating
‘up to what substring it is necessary to read out text
‘returns read out substring
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

‘function settled parameter
‘indicating, after what substring
‘it is necessary to read out the text - the file name
‘and parameter indicated up to what substring
‘it is necessary to read out text
‘then causes the function returning file name
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

‘function settled parameter, indicated
```

```
‘after what substring
‘it is necessary to read out text – start date,
‘and parameter indicated,
‘up to what substring it is necessary to read out text
‘then causes the function returning file name
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

‘function settled parameter, indicated
‘after what substring it is necessary
‘to read out text – start date,
‘and parameter indicated,
‘up to what substring it is necessary to read out text,
‘then causes the function returning file name
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

‘function transforms line value of time to an integer
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
```

After copying of macro text parts to the appropriate sections it is necessary to save the macro project as a separate file. For this purpose from the command menu of editor **File** it is necessary to cause dialog *Save Book*, in which to define file type (**Save as type**) as **Microsoft Excel Add-In** (*.xla) and to save it with necessary name. In a considered example the file with macro has name **Import_OnAirPlayReport.xla**.

4 Use of Macro in *Microsoft Excel*

To connect created macro from application *Microsoft Excel* and to work with it for making of the play backed movies report it is necessary to open dialog *Add-Ins* from menu **Tools** of *Microsoft Excel*. It is necessary to choose from the list **Add-Ins Available** ours macro If in this list necessary macro **Import_Onairplayreport** is absent it is necessary to add it, having pressed on button **Browse…** and having specified in appeared dialog the path to file **Import_OnAirPlayReport.xla**.

After including of macro on the toolbar of application *Microsoft Excel* new button **Import OnAir PlayReport-file** should appear. Pressing this button causes dialog of files select in format **XML** with extension ***.PlayReport**. In this dialog it is necessary to select file name **airMMDD.PlayReport** with the commands list executed by program *FDOnAir* for the accounting period. After loading a file the current sheet of the macro table automatically will rename in **Playback-list** which will contain the information on movies: first column **Date** – playback date, the second **Time** – time of the playback beginning and the third **File** – path to file and its name.

Result of macro work will be the file as table in format *Microsoft Excel* with the playback report list of *FDOnAir* video files with date and time of the beginning of playback.