

## 23.11 Projekte MediaPlayer

In diesem Kapitel werden Ihnen zwei erprobte MediaPlayer-Projekte vorgestellt. Mit dem ersten Programm können Sie Audio-Dateien ausgewählter Formate abspielen, während das zweite Programm einen Video-Player für gängige Video-Formate präsentiert. Beide Projekte verwenden die Klasse MediaPlayer.

### 23.11.1 Projekt 1 - Audio-Player

Das Projekt Audio-Player verfügt über folgende Funktionalität:

- Anzeige des Dateinamens der Audio-Datei
- Audio-Datei abspielen, stoppen und pausieren
- Anzeige der abgelaufenen Abspiel-Zeit
- Vorspulen und Zurückspulen durch einen Klick auf die entsprechende Stelle in der Progressbar
- Anzeige der kompletten Abspiel-Zeit (statisch)
- Auswahl der Audio-Datei in einem Dateiauswahl-Dialog
- Visualisierung der Dynamik des Titels
- Option: Anzeige der Media-Tags Titel, Künstler und Album

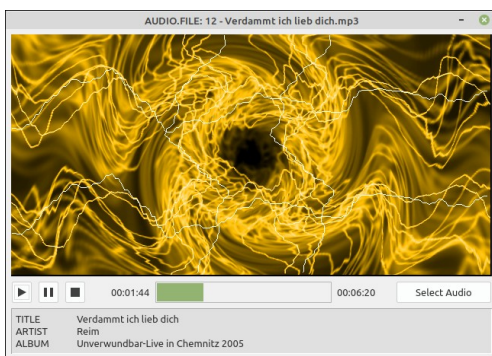


Abbildung 23.11.1.1: Programm-Oberfläche Audio-Player

Die Anzeige der Media-Tags erfolgt nur dann, wenn diese Angaben im Datenstrom vorhanden sind. Im Tag-Event werden die Angaben in die Collection cMetaData eingelesen und später in der Prozedur ShowMetaData() in einem speziellen Fensterbereich angezeigt:

```
Private cMetaData As New Collection
Public Sub hMPlayer_Tag(tagList As MediaTagList)
    Dim sTag As String
    '-- Read selected metadata and save in a collection
    For Each sTag In tagList.Tags
        If sTag = "title" Or sTag = "artist" Or sTag = "album" Then
            If Not cMetaData.Exists(sTag) Then
                cMetaData[Upper(sTag)] = tagList[sTag]
            Endif
        Endif
    Next
End
```

Die Visualisierung der Dynamik des Musikstücks wertet das Programm optisch auf und wird in einer DrawingArea über diese Prozeduren realisiert:

```
Private Sub ShowVisualisation()
    Dim mcVisualisation As MediaControl
    Dim avType As String[] = ["libvisual_infinite", "monoscope", "libvisual_jess",
        "libvisual_jakdaw", "goom", "libvisual_oinksie", "libvisual_bumpscope",
        "spacescope", "spectrascope", "synaescape", "wavescope"]
    '-- Try because distros may not support it
    Try mcVisualisation = New MediaControl(hMPlayer, avType[4])
    If Not Error Then
        hMPlayer.Video.Output = New MediaControl(hMPlayer, "ximagesink")
        hMPlayer.SetWindow(dwaVisualisation)
        hMPlayer.Video.Visualisation = mcVisualisation
        sErrorVisualisation = ""
    End Try
End Sub
```

```

Else
    sErrorVisualisation = "No visualisation available!"
    dwaVisualisation.Refresh()
Endif

End

Public Sub dwaVisualisation_Draw()

    If Not sErrorVisualisation Then
        hImage = Image.Load(Application.Path & "/icons/ft.jpg")
        hImage = hImage.Stretch(dwaVisualisation.W, dwaVisualisation.H)
        Paint.DrawImage(hImage, 0, 0, dwaVisualisation.W, dwaVisualisation.H)
    Else
        Paint.DrawText(sErrorVisualisation, 0, 20)
    Endif

End

```

Den Typ der Visualisierung der Dynamik können Sie statisch aus einer Menge von Typen festlegen:

```

'-- Try because distros may not support it
Try mcVisualisation = New MediaControl(hMPlayer, avType[4])

```

### 23.11.2 Projekt 2 - Video-Player

Das Projekt Video-Player verfügt über folgende Funktionalität:

- Anzeige des Dateinamens der Video-Datei
- Video-Datei abspielen, stoppen und pausieren
- Anzeige der abgelaufenen Abspiel-Zeit
- Vorspulen und Zurückspulen durch Änderung der Position des Sliders
- Anzeige der kompletten Abspiel-Zeit (statisch)
- Änderung der Lautstärke
- Änderung von Video-Einstellungen wie zum Beispiel Helligkeit oder Kontrast in einem Dialog
- Vollbild-Modus
- Auswahl der Video-Datei in einem Dateiauswahl-Dialog
- Automatisches Drehen eines Videos

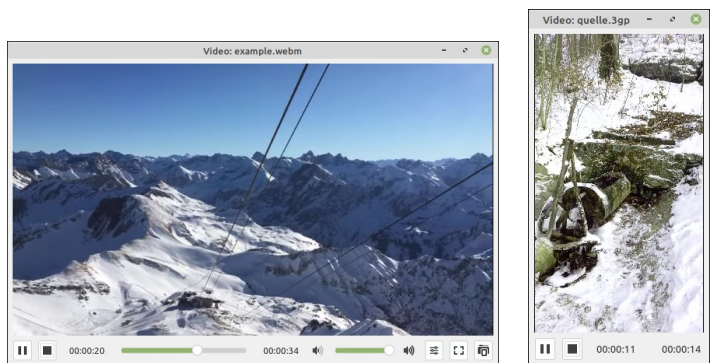


Abbildung 23.11.2.1: Programm-Oberfläche Video-Player

Eine Besonderheit im vorgestellten Video-Player zeigt sich bei der Anzeige eines Videos. Unabhängig vom Originalformat wird das Video automatisch stets so gedreht, dass es korrekt angezeigt wird. Die Notwendigkeit der Drehung zeigte sich zum Beispiel bei Video-Aufnahmen von einem Mobiltelefon.

Nach einem Auswahl-Dialog wird in der privaten Prozedur Start() die Ausgabe des Video-Media-Players mit dem MediaContainer verbunden und dieser anstelle der Standardausgabe verwendet. Als Flip-Methode wird aus der Menge der Methoden 'automatic' eingesetzt, die stets eine korrekte Anzeige ergab:

```

Private Enum none = 0, clockwise, rotate_180, counterclockwise, horizontal_flip, vertical_flip,
    upper_left_diagonal, upper_right_diagonal, automatic

Private MContainer As MediaContainer
Private MCFlip As MediaControl
Private MCSink As MediaControl

Public Sub btnFileOpen_Click()

    Dialog.Title = ("Select a video file ...")
    Dialog.Filter = ["*.avi;*.mp4;*.webm;*.mov;*.mkv;*.3gp;*.ts", "Video files", "*", "All files"]

```

```
Dialog.Path = Desktop.GetDirectory("VIDEOS") & "/"
Dialog.ShowHidden = False

If Dialog.OpenFile(False) Then Return

Me.Text = "Video: " & File.Name(Dialog.Path)

Start()

End

Private Sub Start()

    MPlayer = Null

'-- The media-objects have to be newly generated before playing because of videoflip,
'-- which only works correctly one time after initialization.
MPlayer = New MediaPlayer As "MPlayer"
MContainer = New MediaContainer(MPlayer)
MCFliP = New MediaControl(MContainer, "videoflip")
MCFliP["method"] = automatic
MCSink = New MediaControl(MContainer, "xvimagesink")
MCFliP.LinkTo(MCSink)
MContainer.AddInput(MCFliP, "sink")

'-- Connect output of mediaPlayer with MediaContainer and use it instead of the default output
MPlayer.Video.Output = MContainer

MPlayer.URL = Media.URL(Dialog.Path)
MCSink.SetWindow(DrawingArea1)
Slider1.Value = 0

Try MPlayer.Play()
If Error Then
    Message.Error("An error occured.\nCan't play selected video file.")
    Return
Else
    ...
Endif

End
```

Im Download-Bereich werden Ihnen beide Projekte jeweils als Quelltext-Archiv zur Verfügung gestellt.