

Scripting

Scripting funktioniert nur bei der 4K oder PRO Version.

Viele Beispiele und Tricks für die folgenden Scripts sind aus „doggy's“ [SCRIPTING FOR DUMMIES](#) und anderen Beiträgen vom [vMix Forum](#).

Scripting ist innerhalb von vMix mit dem eingebauten Editor möglich oder als externer Befehl via Browser. Die Zusammenstellung der Beispiele ist unvollständig. Im Anhang hat es eine Auflistung der zurzeit bekannten API-Befehle. vMix Scripting unterstützt VB.NET beim Schreiben eines Skripts. Alle Variablen sind Lokal, es gibt leider keine Public Variablen. Man kann irgendwelche nicht sichtbaren Textfelder als Public Variablen zweckentfremden.

Bei Input kann entweder der Filename des Titels (Gross-Kleinschreibung beachten) oder die InputNummer verwendet werden.

vMix Script Builder

Am Anfang ist der vMix Script Builder (ein externes Hilfsprogramm) eine unverzichtbare Hilfe. Erstellt Scripts mit der korrekten Syntax, Auswahl via Pulldown Menus. Auch aus dem vMix Forum, Universal Title Controller stammt vom gleichen Autor.

[vMix Script Builder](#)

vMix Script Editor

Die Beschreibung auf der vMix Seite ist etwas dürftig, trotzdem lohnt sich immer ein Blick dorthin.

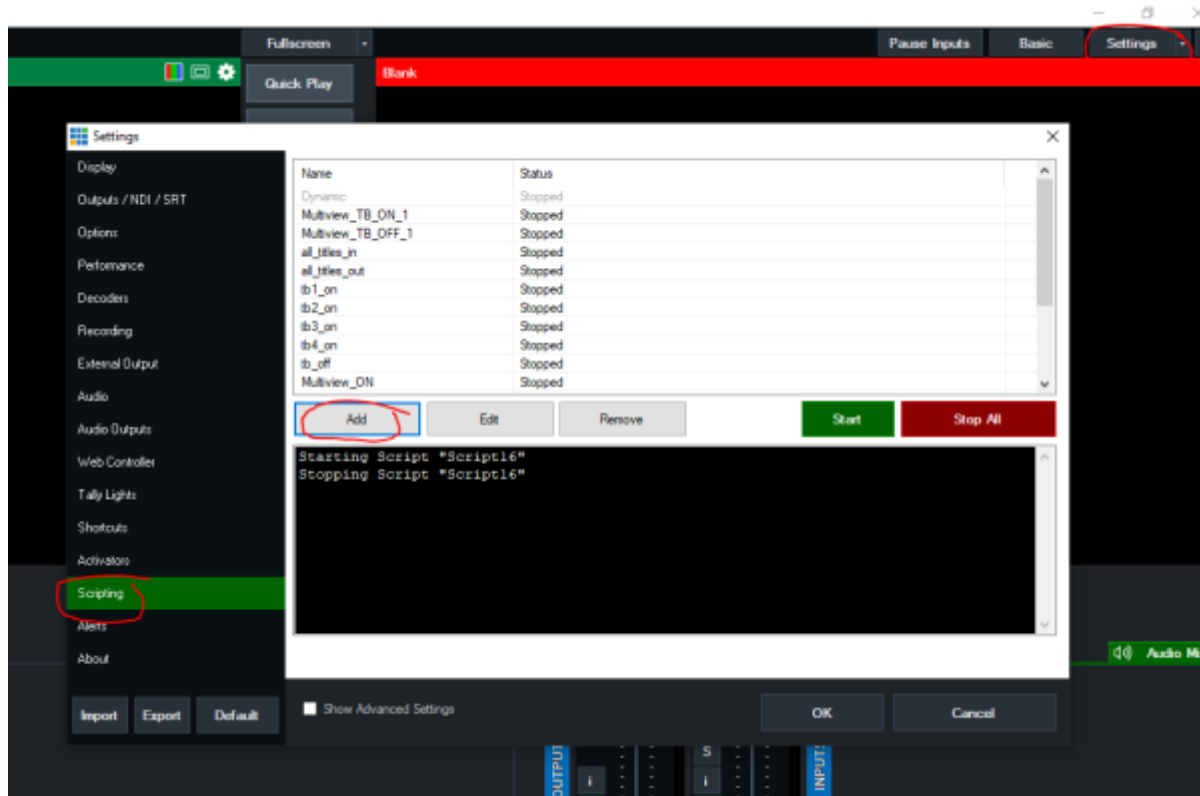
[vMix API Website](#)

Hier die aktuell gültige Liste der Befehle in der Übersicht.

[Shortcut Version 23 Übersicht](#)

[Shortcut Version 24 Übersicht](#)

Den Scripteditor erreicht man via „Settings“, „Scripting“ Add



VERMEIDE LEERZEICHEN im SCRIPTNAMEN

Ein Script kann via einen Shortcut abgerufen werden.

Add Shortcut [X]

General
Advanced

Key/Control: A [v] [Find ...]

Function: ScriptStart [v]

Value: script16 [v]
Script Name

Title: []

Description1: []

Description2: []

Display: Default [v]

☐ Local Shortcut (This Preset Only)
☒ Show in Web Controller

OK **Cancel**

4 verschiedene Arten, ein script zu schreiben

es gibt 4 verschiedene Arten ein script zu schreiben. Schlussendlich machen die Varianten alle das selbe. Im normalfall reicht in vMix Editor die einfachste Variante. (WEB)

- VB.net kann aber mit variablen umgehen, WEB kann das nicht.
- API entspricht VB net
- WEB ohne variablen, kurz und einfach, reicht für die meisten internen scripte.
- HTTP sendet den API Befehl via Browser oder einer externen Software.

Es funktioniert sowohl AudioBusOFF, als auch audiobussoff.

ACHTUNG: Titel und Textfelder beachten die Gross-Kleinschreibung:

Title 0- The Classic Blue.gzip ist genau wie der Titel heisst. Bei Aenderung der Gross-Kleinschreibung funktionier das Script nicht.

Dieses Beispiel schreibt „Some Text“ in ein bestimmtes Textfeld eines bestimmten Titels. Anstelle des Titelnamens kann auch die Input Nummer spezifiziert werden, anstelle des Textfeldes, dessen Indexnummer.

VB.net

```
Input.Find("Title 0- The Classic Blue.gtzip").Text("Headline.Text")= "Some Text"
```

API

```
API.Function("SetText",Input:"1",SelectedName:"Headline.Text",Value:"Some Text")
```

WEB

```
Function=SetText&Input=1&SelectedIndex=0&Value=Some Text
```

HTTP

```
http://127.0.0.1:8088/api/?Function=SetText&Input=Title 0- The Classic Blue.gtzip&SelectedName=Headline.Text&Value=Some Text
```

API Daten aus vMix auslesen via HTTP

Damit erhältst Du alle verfügbaren Daten der API (Application Programming Interface) von vMix, als XML zurück.

```
http://127.0.0.1:8088/api/?
```

Achtung bei der codierung von Sonderzeichen bei URL's (API via HTTP)

Sei vorsichtig, wenn Du Webadressen mit Symbolen wie , # usw. verwendest, da diese korrekt „kodiert“ sein müssen, um als Url-Wert zu funktionieren. Dies ist nicht nur bei vMix der Fall, sondern bei allen webbasierten APIs. Dies ist ein praktisches Tool, das verwendet werden kann, um die Werte korrekt zu kodieren: [URL Decoder/Encoder](#)

Gib nur den Text ein, den Du als Wert setzen willst, nicht die ganze Url. So würde aus

```
Excel/CSV,Sheet1,5
```

folgendes werden

```
Excel%2FCSV%2CSheet1%2C5
```

Und die ganze URL wiederum wäre:

```
http://localhost:8088/API/?Function=DataSourceSelectRow&Value=Excel%2FCSV%2CSheet1%2C5
```

Hello World Beispielscript

Dieses Beispiel verwendet den mitgelieferten Titel *NewsHD* und schreibt Hello World in das Feld *Headline*

```
Input.Find("NewsHD.xaml").Text("Headline") = "Hello World!"
```

oder anders ausgedrückt:

```
dim i = Input.Find("NewsHD.xaml")  
i.Text("Headline") = "Hello World!"
```

Output (ändern)

Eingang 4 Video auf OUTPUT routen

```
Function=Activeinput&Input=4  
oder  
Function=Cutdirect&Input=4  
oder  
http://127.0.0.1:8088/API/?Function=Cutdirect&Input=4  
oder  
API.Function("Cutdirect",4,)
```

Eingang 3 Video auf Preview routen

```
Function=Previewinput&Input=4
```

Eingang 1 Video auf **OUTPUT2** routen

```
Function=SetOutput2&Input=1&Value=1
```

Eingang mit einem bestimmten Namen auf die Linie routen

```
Input.Find("YourInput").Function("Cut")
```

Input

Schneidet Input 4 zu PGM (Cut)

Cut schneidet den Input 4 auf den Ausgang und wechselt den vorherigen Input auf Preview, Cudirect belässt den vorgewählten Preview-Input.

```
API.Function("Cut",4,)
```

oder

```
API.Function("Cutdirect",4,)
```

Schneidet Input 4 zu PREVIEW

```
API.Function("PreviewInput",4,)
```

Fade Input (oder jede Art von transition) zu PGM

Mix 0 = PGM

MIX 1-3 = Input Mixes, nur bei den teureren Versionen verfügbar.

-1 = PREVIEW

fade Preview auf PGM

```
API.Function("Fade",Input:=-1,Mix:=1)
```

cut Input 2 auf PGM

```
API.Function("Fade",Input:=2,Mix:=1)
```

add Input per code

Alle diese Funktionen erzeugen einen neuen Eingang

Titel Input von einem bestimmten Verzeichnis

```
http://127.0.0.1:8088/API/?Function=AddInput&Value=Xaml|C:\VMIX\Name.gtzip
```

Video Input von einem bestimmten Verzeichnis

```
http://127.0.0.1:8088/API/?Function=AddInput&Value=Video|c:\path\to\video.avi
```

Image Input von einem bestimmten Verzeichnis

```
http://127.0.0.1:8088/API/?Function=AddInput&Value=Image|c:\path\to\image.jpg
```

Picture Folder Input von einem bestimmten Verzeichnis

```
http://127.0.0.1:8088/API/?Function=AddInput&Value=Photos|c:\path\to\folder
```

Video Playlist (m3u) Input von einem bestimmten Verzeichnis

```
http://127.0.0.1:8088/API/?Function=AddInput&Value=VideoList|c:\path\to\playlist.m3u
```

Audio file Input von einem bestimmten Verzeichnis

```
http://127.0.0.1:8088/API/?Function=AddInput&Value=AudioFile|c:\path\to\audio.wav
```

Powerpoint Input von einem bestimmten Verzeichnis

(es funktionieren nur Folien ohne Videos oder Audios bzw. Mediafiles werden nicht gespielt)

```
http://127.0.0.1:8088/API/?Function=AddInput&Value=PowerPoint|c:\path\to\powerpoint.pptx
```

Color Input (RGB Farbwert)

```
http://127.0.0.1:8088/API/?Function=AddInput&Value=Colour|%23FF5733
```

tauscht zwei Inputs hin und her

damit können, z.B. via Leertaste, zwei Inputs hin und her getauscht werden. Der Inputname vom ersten Input dient als Abfrage

```
if word = "DeckLink Duo (1) 1" then
API.Function("MoveInput",Input:="DeckLink Duo (1) 1" , Value:=2)
else
API.Function("MoveInput",Input:="DeckLink Duo (1) 1" , Value:=1)
end if
```

Audio

Audio Bus der Quelle 1 Einschalten (grüner Lausprecher)

```
Function=AudioON&Input=1
oder
API.Function("AudioON",1,)
```

Audio Bus der Quelle 1 auf Audio Ausgang D routen

```
Function=AudioBusON&Input=1&Value=D
oder
API.Function("AudioBusOff",2,"A")
```

Audio Bus der Quelle 2 auf Audio Ausgang Master routen

```
Function=AudioBusON&Input=2&Value=M
```

Audio Bus der Quelle 1 vom Master wegrouten

```
Function=AudioBusOFF&Input=1&Value=Master
```


Audio Bus der Quelle 3 vom D wegrouten

```
Function=AudioBusOFF&Input=1&Value=D
```

Audio follow Video vom Eingang 1 abschalten

```
Function=AudioAutoOff&Input=1
```

Schaltet bei allen Eingängen den AudioBus D aus

```
Dim doc As New XmlDocument()
doc.LoadXml(API.Xml)
Dim root As XmlNode = doc.DocumentElement
Dim i as Integer
Dim xel As XmlElement
Dim node As XmlNode = root.SelectSingleNode ("/vmix/active") 'look for node
Dim inputNumber as Integer = node.InnerXml 'shows active input

'wieviele Eingänge sind aktiv
Dim vmixinputs as Integer 'inputs in total
For Each xel In Doc.SelectNodes("vmix")
    vmixinputs = (xel.SelectNodes("inputs/input").Count)
Next xel

for i = 1 to vmixinputs
    API.Function("AudioBusOff", i, "D")
Next
```

Holt Audio Titelname aus einem Input

Dieses Skript liest den aktuellen Titel einer Wiedergabeliste und schreibt ihn formatiert in ein Feld eines Titels.

In diesem Beispiel verwende ich den in vMix enthaltenen Titel „ Title 33- On the shelf- Peach.gzip “ und schreibe den Text in das Feld „Headline.Text“. Mein List-Input ist Input2. Bitte passe diese (Titel, Feld und List-Input-Nummer) für dein Projekt an.

Ich verwende eine Liste von MP3-Audiodateien. Diese sind wie folgt benannt:

Anni Piper - Two's Company - 01 - Blues Before Sunrise.mp3

Anni Piper - Two's Company - 02 - Live To Play.mp3

Anni Piper - Two's Company - 03 - Man's Woman.mp3

Das Ziel ist es, aus dieser Zeichenkette einen brauchbaren Namen zu extrahieren.

Das fertige Skript sieht wie folgt aus:

```
'script playing now
do
dim xml as string = API.XML()
dim x as new system.xml.xmldocument
x.loadxml(xml)
dim word as string =
x.SelectSingleNode("//input[@number=[h]'2'[/h]]/@title").Value
word = word.remove(word.Length - 4)
dim wordArr as String() = word.Split("-")
Dim result0 as String = wordArr(0)
Dim result1 as String = wordArr(1)
Dim result2 as String = wordArr(2)
Dim result3 as String = wordArr(3)
Dim result4 as String = wordArr(4)
dim fullname as string = result1 + " " + result2 + " " + result4
API.Function("SetText",Input:="Title 33- On the shelf-
Peach.gtzip",SelectedName:="Headline.Text",Value:=result1)
sleep(1000)
API.Function("SetText",Input:="Title 33- On the shelf-
Peach.gtzip",SelectedName:="Headline.Text",Value:=result2)
sleep(1000)
API.Function("SetText",Input:="Title 33- On the shelf-
Peach.gtzip",SelectedName:="Headline.Text",Value:=result4)
sleep(1000)
API.Function("SetText",Input:="Title 33- On the shelf-
Peach.gtzip",SelectedName:="Headline.Text",Value:=fullname)
sleep(1000)
loop
```

konvertieren eines API XML Wertes in einen Lautsärkewert eines Faders

Der via API ausgelesene Lautstärkewert eines Inputs entspricht kann nicht 1:1 auf ein Fadervolume eines Ausgangs gesetzt werden, Der Wert muss konvertiert werden.

```
dim x as new system.xml.xmldocument ' Create the XML document to parse
x.loadxml(API.XML()) ' Load the API into the XML document. For a visual, see
http://127.0.0.1:8088/api

' Holt den aktuellen Wert Attribute "Volume" vom Input(in diesem Fall
Input1)
dim xmlVolume As Double =
x.SelectSingleNode("//input[@number='1']/@volume").Value

xmlVolume = xmlVolume / 100
'Vermutlich ist dies eigentlich die Amplitude und nicht Volume. Es wird
```

erwartet, dass Amplitude in der Formel ein normalisierter (0-1) Wert ist, was in der XML nicht der Fall ist.

```
dim faderVolume As Integer = cint((xmlVolume ^ 0.25) * 100) ' Formula:
Volume = (Amplitude ^ 0.25) * 100
```

```
'nun setzen wir die Lautstärke von BUS A auf die Lautstärke des Inputs 1
API.Function("SetBusAVolume",,Value:=faderVolume )
```

Audio Bus Status von allen Inputs

Zeigt in einem Titel an, welche Audio-Busse von den Inputs belegt sind. Titel hier:

audiochannels.gtzip

Die Ausgabe ist nicht sehr elegant, der Titel löscht zwischendurch ganz, zeigt aber die Möglichkeiten auf.

```
do while true

'löscht titel
Input.Find("audiochannels.gtzip").Text("Text1.Text")=""

'wieviele Eingänge sind aktiv
dim doc As New XmlDocument()
doc.LoadXml(API.Xml)
dim root As XmlNode = doc.DocumentElement
dim xel As XmlElement
dim node As XmlNode = root.SelectSingleNode("/vmix/active") 'look for node
Dim vmixinputs as Integer 'inputs insgesamt
For Each xel In Doc.SelectNodes("vmix")
    vmixinputs = (xel.SelectNodes("inputs/input").Count)
Next xel

' loop zum auslesen der XML API auf Audioeingänge
Dim i as integer
dim eingang as string
dim text as string
dim titel as string
for i = 1 to vmixinputs
    try
        dim xml as string = API.XML()
        dim x as new system.xml.xmldocument
        x.loadxml(xml)
        eingang = (x.SelectSingleNode("/vmix/inputs/input[" & cstr(i)
&"]/@audiobusses").InnerText)
        titel = (x.SelectSingleNode("//input[@number="& cstr(i) + "]").InnerText)
        console.writeline(cstr(i) & " " & eingang)
        text = Input.Find("audiochannels.gtzip").Text("Text1.Text")
        eingang = eingang.padright(25," ")
```

```

    Input.Find("audiochannels.gtzip").Text("Text1.Text")=text +
Environment.NewLine + cstr(i)+ " " + eingang + " " + titel
Catch ex as Exception
    console.writeline(cstr(i)+ " does not contain Audio")
    text = Input.Find("audiochannels.gtzip").Text("Text1.Text")
    Input.Find("audiochannels.gtzip").Text("Text1.Text")=text +
Environment.NewLine + cstr(i) + " does not contain Audio"
end try
next
sleep (200)
loop

```

welcher Audiobus ist auf einem bestimmten Input online?

```

dim audios as string

do while true

Dim xml as string = API.XML()
Dim x as new system.xml.xmldocument
x.load("http://localhost:8088/api")
audios= (x.SelectSingleNode("//input[1]/@audiobusses").InnerText)
console.writeline(eingang )

If audios= "M,A"
'do this
Else if audios= "M,A,B"
'do that
else
'do something else
end if

loop

```

Titel / Text

Ändert Text in einem bestimmten Feld

```

Input.Find("Title 0- The Classic Blue.gtzip").Text("Headline.Text")= "Text
geändert"
oder
API.Function("SetText",Input:="Title 0- The Classic
Blue.gtzip",SelectedName:="Headline.Text",Value:="Text geändert")
oder
http://127.0.0.1:8088/api/?Function=SetText&Input=Title 0- The Classic

```

Blue.gtzip&SelectedName=Headline.Text&Value=Text geändert

Ändert Schrift Farbe eines bestimmten Textfeldes

Es gehen argb Werte oder gültige Namen für eine Farbe (red, blue, black, transparent etc.) argb Werte und Namen findest Du hier https://www.w3schools.com/colors/colors_names.asp

```
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg21.Text",Value:="transparent")
```

Zahlenwert auf Tastendruck erhöhen, verringern

Addieren +=1

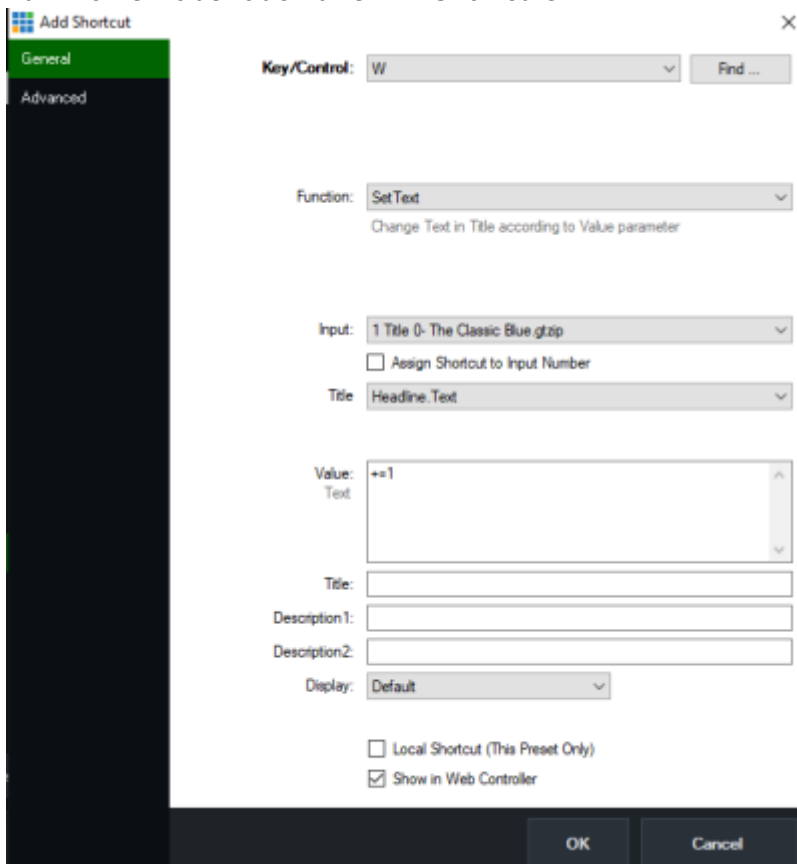
Subtrahieren -=1

Das Script einer Taste zuordnen.

```
Input.Find("Title 0- The Classic Blue.gtzip").Text("Headline.Text")= "+=1"
oder
```

```
API.Function("SetText",Input:="Title 0- The Classic
Blue.gtzip",SelectedName:=" Headline.Text ",Value:="+=1")
```

Funktioniert aber auch direkt in Shortcuts.



Titel / Text via browser

Als IP Adresse benutzt man folgendes:

- 127.0.0.1 wenn der Browser auf demselben Computer läuft, wie vMix.
- die IP des anderen vMix Rechners, im selben Netzwerk

Ändert ein (Logo, Foto etc.)

```
http://127.0.0.1:8088/API/?Function=SetImage&Input=Titel.gtzip"&SelectedName=LOGO.Source&Value=C:\VMIX\ch.png
```

Setzt ein Bild (Logo, Foto etc.) auf unsichtbar (sichtbar SetImageVisibleOn)

```
http://127.0.0.1:8088/API/?Function=SetImage&Input=Titel.gtzip&Image=LOGO.Source&Value=C:\VMIX\ch.png
```

Ändert Text in einem bestimmten Feld via Browser

```
http://127.0.0.1:8088/api/?Function=SetText&Input=Title 0- The Classic Blue.gtzip&SelectedName=Headline.Text&Value=Text geändert
```

Ändert Textfarbe in einem bestimmten Feld via Browser

```
http://127.0.0.1:8088/api/?Function=SetTextColour&Input=Title 0- The Classic Blue.gtzip&SelectedName=Headline.Text&Value=red
```

komplexeres Beispiel, ändert Titel Texte, je nachdem welcher Input aktiv ist

in welchem zuerst die aktive Input Nummer festgestellt wird, dann aufgrund der Inputnummer eine Aktion erfolgt. Es läuft als Loop, also solange jemand das Script startet und nicht wieder anhält.

```
do while true

Dim doc As New XmlDocument()
doc.LoadXml(API.Xml)
Dim root As XmlNode = doc.DocumentElement

Dim node As XmlNode = root.SelectSingleNode("/vmix/active")
Dim inputNumber as Integer = node.InnerXml
```

```
If inputNumber = 2 Then
Input.Find("whoisonair.gtzip").Text("Text11.Text") = "caller1 onair OUT1"
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg11.T
ext",Value:="red")
Input.Find("whoisonair.gtzip").Text("Text21.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text31.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text41.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text51.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text61.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text71.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text71.Text") = " "
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg21.T
ext",Value:="transparent")
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg31.T
ext",Value:="transparent")
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg41.T
ext",Value:="transparent")
End If
```

```
If inputNumber = 3 Then
Input.Find("whoisonair.gtzip").Text("Text21.Text") = "caller2 onair OUT1"
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg21.T
ext",Value:="red")
Input.Find("whoisonair.gtzip").Text("Text11.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text31.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text41.Text") = " "
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg11.T
ext",Value:="transparent")
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg31.T
ext",Value:="transparent")
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg41.T
ext",Value:="transparent")
End If
```

```
If inputNumber = 4 Then
Input.Find("whoisonair.gtzip").Text("Text31.Text") = "caller3 onair OUT1"
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg31.T
ext",Value:="red")
Input.Find("whoisonair.gtzip").Text("Text11.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text21.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text41.Text") = " "
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg11.T
ext",Value:="transparent")
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg21.T
ext",Value:="transparent")
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg41.T
ext",Value:="transparent")
End If
```

```
If inputNumber = 5 Then
Input.Find("whoisonair.gtzip").Text("Text41.Text") = "caller4 onair OUT1"
```

```
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg41.Text",Value:="red")
Input.Find("whoisonair.gtzip").Text("Text11.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text21.Text") = " "
Input.Find("whoisonair.gtzip").Text("Text31.Text") = " "
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg11.Text",Value:="transparent")
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg21.Text",Value:="transparent")
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg31.Text",Value:="transparent")
End If
sleep (200)
loop
```

Uhrzeit für eine anderen Zeitzone

Die Zahl in der Variablen AddHours(3), bestimmt die andere Zeitzone von der des laufenden Computers, auf welchem das script läuft. Der Titel heisst uhr.gtzip, es wird ins erste Feld des Titels geschrieben (änderbar mit der Indexnummer)

```
do while true
Dim d1 As DateTime = DateTime.Now.AddHours(3)
API.Function("SetText",Input:="uhr",SelectedIndex:="0",Value:=d1.ToString("dd/MM/yy H:mm:ss"))
sleep(200)
loop
```

Diverses

Wartet 1 Sekunde

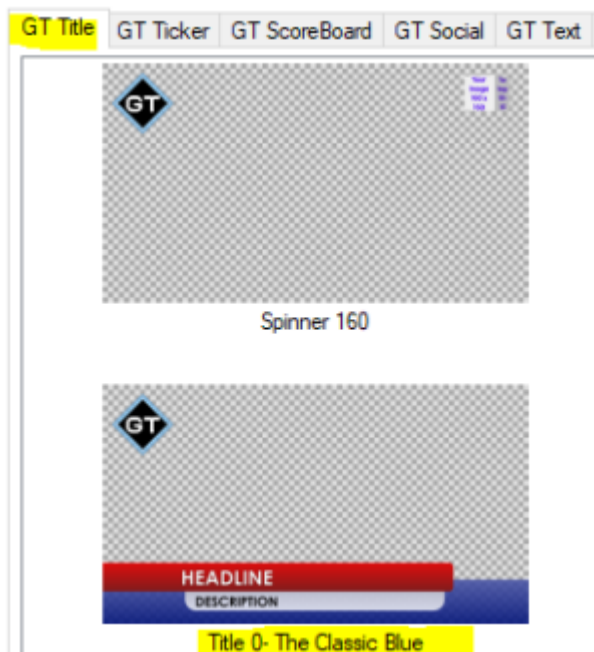
```
sleep(1000)
```


Holt den Names eines vMix Callers vom Input 1

```
do while true
dim xml as string = API.XML()
dim x as new system.xml.xmldocument
x.loadxml(xml)
dim word as string = (x.SelectSingleNode("//input[@number=1]").InnerText)
console.writeline(word) 'or API to send to title
sleep(500)
loop
```

läuft die Aufzeichnung?

Aufzeichnungsstatus via API abfragen und als Anzeige den Wert in einem Title Input schreiben. Als Titel ist ein Beispieltitel aus der GT Library gewählt. **Title 0- The Classic Blue.gtzip**



```
dim isrecording as string = ""

' do while true/loop prüft stetig, bis das script angehalten wird
do while true

    dim xml as string = API.XML()
    dim x as new system.xml.xmldocument
    x.loadxml(xml)

'sind wir am aufzeichnen?
    isrecording = (x.SelectSingleNode("/vmix/recording").InnerText)
'eine Antwort in einen Titel schreiben und die Farbe entsprechend ändern
if isrecording = true
    API.Function("SetText",Input:="Title 0- The Classic
Blue.gtzip",SelectedName:="Headline.Text",Value:="AUFZEICHNUNG LÄUFT")
```

```

API.Function("SetTextColour",Input:="Title 0- The Classic
Blue.gtzip",SelectedName:="Headline.Text",Value:="red")
else
API.Function("SetText",Input:="Title 0- The Classic
Blue.gtzip",SelectedName:="Headline.Text",Value:="AUFZEICHNUNG STOP")
API.Function("SetTextColour",Input:="Title 0- The Classic
Blue.gtzip",SelectedName:="Headline.Text",Value:="green")
end if

sleep(500)
loop

```

Zeigt die verbleibende Zeit des laufenden Videos in einem Titel an

Titelbeispiel:

remainingtime.gtzip

```

' Die verbleibende Zeit des aktiven laufenden Videos prüfen und in einem
Titel anzeigen.

```

```

dim position as string = ""
dim duration as string = ""
dim activeinput as string = ""
dim Timeleft as double = 0
dim triggertime as integer = 10 '10 Sekunden vor Ende

```

```

do while true
dim xml as string = API.XML()
dim x as new system.xml.xmldocument
x.loadxml(xml)

```

```

activeinput = (x.SelectSingleNode("//active").InnerText)
duration = (x.SelectSingleNode("//input[@number='"& activeinput
&"']/@duration").Value)
position = (x.SelectSingleNode("//input[@number='"& activeinput
&"']/@position").Value)

```

```

Timeleft= Double.Parse(duration)-Double.Parse(position)
Timeleft = Timeleft / 100
dim Timingleft as integer = CInt(Timeleft)
Timingleft = Timingleft / 10
dim Minutes as integer = Timingleft \ 60
dim Seconds as integer = Timingleft Mod 60

```

```

'zum testen ausklammern
' console.writeline(Timingleft)
' console.writeline(Seconds)

```

```

'editiere den Text "verbleibende Zeit" nach Bedarf
dim ThisTime as string
ThisTime = "verbleibende Zeit - " + Minutes.ToString("00") + ":" +
Seconds.ToString("00")

if Timingleft < 60
    dim TimeRemaining as string
    TimeRemaining = "remaining time - 00:" + Seconds.ToString("00")
    API.Function("SetText", Input:="remainingtime.gtzip",
SelectedIndex:="0", Value:=Timerremaining)
    if Timingleft < 30
API.Function("SetTextColour",Input:="remainingtime.gtzip",Value:="red")
    else
API.Function("SetTextColour",Input:="remainingtime.gtzip",Value:="orange")
    end if
else
    API.Function("SetText",Input:="remainingtime.gtzip",SelectedIndex:="0"
,Value:=ThisTime)
    API.Function("SetTextColour",Input:="remainingtime.gtzip",Value:="green")
end if
sleep(50)
Loop

```

ein Ereignis auslösen, z.B. 2 Sekunden bevor das Video fertig ist?

Dieses Beispiel blendet den Ton des Videos 2 Sekunden(triggertime) bevor das Video fertig ist langsam(triggerduration)aus.

```

dim position as string = ""
dim duration as string = ""
dim activeinput as string = ""
dim Timeleft as double = 0
dim triggertime as integer = 2000      '2 Sekunden bevor Ende
dim triggerduration as integer = 500   'Ausblendzeit (fade)

do while true

    dim xml as string = API.XML()
    dim x as new system.xml.xmldocument
    x.loadxml(xml)

    activeinput = (x.SelectSingleNode("//active").InnerText)
    duration = (x.SelectSingleNode("//input[@number='"& activeinput
&"']/@duration").Value)
    position = (x.SelectSingleNode("//input[@number='"& activeinput
&"']/@position").Value)

    Timeleft= Double.Parse(duration)-Double.Parse(position)

```

```
if Timeleft < triggertime
  API.Function("SetVolumeFade",Input:=activeinput.toString(),Value:="0," &
triggerduration.toString())
  ' hier tun, was man tun möchte, das Beispiel blendet den Ton langsam aus
end if

sleep(500)
Loop
```

Video auf eine bestimmte Stelle positionieren

```
http://127.0.0.1:8088/api/?Function=SetPosition&Input=Video.mp4&Value=2000
```

Video starten

Input 0 ist PGM

```
"Function=Play&Input=1")
```

zählt 2 Zahlen aus 2 Textfeldern zusammen und zeigt das Ergebnis in einem 3. Textfeld an

```
Dim one as integer =
Convert.ToInt32(Input.Find("Tally.gtzip").Text("One.Text"))
Dim Two as integer =
Convert.ToInt32(Input.Find("Tally.gtzip").Text("Two.Text"))

API.Function("SetText",Input:="Tally.gtzip",SelectedName:="Total.Text",Value
:=(One + Two).toString)
```

schreibt einen Zeitstempel in eine Textdatei

Während eine Aufnahme kann der Redaktor eine Taste drücken, welche das Script auslöst. Es wird ein Zeitstempel in ein Textfile geschrieben, dies erleichtert das Auffinden von gewünschten Stellen in der Aufzeichnung.

```
'Text file für die Zeitmarken
dim FILENAME = "D:\out.txt"

'Get recording duration
dim xml = API.XML()
dim x as new system.xml.xmldocument
x.loadxml(xml)
```

```

dim Duration =
integer.parse((x.SelectSingleNode("//recording/@duration").Value))

dim Second = math.floor(Duration mod 60)
dim Minute = math.floor((Duration / 60) mod 60)
dim Hour = math.floor((Duration / (60 * 60)) mod 60)

'Aufnahmezeit in die Datei schreiben
System.IO.File.AppendAllText(filename,
string.format("{0:00}:{1:00}:{2:00}"&Environment.NewLine, Hour, Minute,
Second))

```

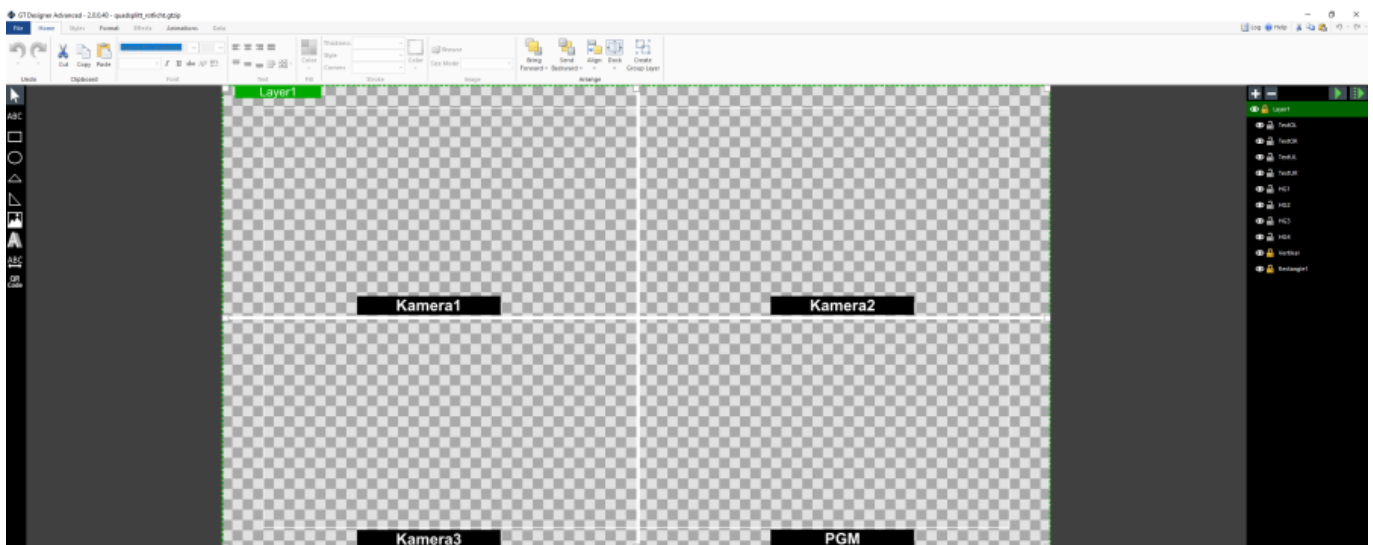
Quadsplitt Rotlicht

Es muss ein Multiviewer Quadsplitt vorhanden sein, mit einem Titel Overlay(Quadsplitt_Rotlicht.gtzip).

download quadsplitt_rotlicht.gtzip GT Template

Da sich in vMix farbige Rechtecke eines Titels nicht umfärben lassen, Text aber schon, verwende ich das ARIAL ASCII Zeichen U+2588, Voller Block.█

Das Text-Zeichen lässt sich in der Farbe einfach anpassen.



```

do while true

Dim doc As New XmlDocument()
doc.LoadXml(API.Xml)
Dim root As XmlNode = doc.DocumentElement

Dim node As XmlNode = root.SelectSingleNode("/vmix/active")
Dim inputNumber as Integer = node.InnerXml

If inputNumber = 1 Then
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("Text0L.Text") = "Kamera 1
ONAIR"

```

```
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("Text0R.Text") = "Kamera 2"
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("TextUL.Text") = "Kamera 3"
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("TextUR.Text") = "PGM K1"
API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedName
:="HG1.Text",Value:="red")
API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedName
:="HG2.Text",Value:="black")
API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedName
:="HG3.Text",Value:="black")
'API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedNam
e:="HG4.Text",Value:="black")
end if

If inputNumber = 2 Then
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("Text0L.Text") = "Kamera 1"
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("Text0R.Text") = "Kamera 2
ONAIR"
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("TextUL.Text") = "Kamera 3"
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("TextUR.Text") = "PGM K2"
API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedName
:="HG1.Text",Value:="black")
API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedName
:="HG2.Text",Value:="red")
API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedName
:="HG3.Text",Value:="black")
'API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedNam
e:="HG4.Text",Value:="black")
end if

If inputNumber = 3 Then
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("Text0L.Text") = "Kamera 1"
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("Text0R.Text") = "Kamera 2"
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("TextUL.Text") = "Kamera 3
ONAIR"
Input.Find("Quadsplitt_Rotlicht.gtzip").Text("TextUR.Text") = "PGM K3"
API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedName
:="HG1.Text",Value:="black")
API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedName
:="HG2.Text",Value:="black")
API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedName
:="HG3.Text",Value:="red")
'API.Function("SetTextColour",Input:="Quadsplitt_Rotlicht.gtzip",SelectedNam
e:="HG4.Text",Value:="black")
end if

sleep (200)
loop
```

String formatieren

vMix macht manchmal aus einer Zahl, welche von einer Datenquelle stammt, etwas „anderes“.
aus 100.00 wird 100
aus 100.10 wird 100.1
etc.

Falls aber immer die Anzeige 100.00 gewünscht ist, kann das via ein loopendes Script korrigiert werden. In diesem Beispiel ist es der Titel mytitle und das Feld Headline.Text.

```
do
dim j as integer
dim i As string = ""
dim mystring as string = ""

i = Input.Find("mytitle").Text("Headline.Text")
j = CInt(i)

myString = String.Format("{0:00}", j)

API.Function("SetText",Input:="mytitle",SelectedName:="Description.Text",Value:=myString )
loop
```

Multiviewer einschalten und belegen

Schaltet auf den ersten 4 Inputs den ersten Multiviewer Layer ein und belegt diese mit Inputs 11-14

```
Function=SetMultiViewOverlay&Input=1&Value=1,11
Function=MultiViewOverlayON&Input=1&Value=1

Function=SetMultiViewOverlay&Input=2&Value=1,12
Function=MultiViewOverlayON&Input=2&Value=1

Function=SetMultiViewOverlay&Input=3&Value=1,13
Function=MultiViewOverlayON&Input=3&Value=1

Function=SetMultiViewOverlay&Input=4&Value=1,14
Function=MultiViewOverlayON&Input=4&Value=1
```

oder

```
API.Function("SetMultiViewOverlay",Input:="1",Value:="1,11")
API.Function("MultiViewOverlayOFF",Input:="1",Value:="1")
```

Script aus Script starten

ACHTUNG: das Aufrufen eines Skriptes aus einem Script braucht Zeit. Also immer nach einer komplexeren Aktion ein `sleep(200)` setzen.

```
API.Function("ScriptStart", , "MeinScript")
sleep(200)
```

Script, welches einen Befehl über das Netzwerk an einen anderen vMix Computer schickt

Dieses Beispiel schaltet Input1 auf Overlay1 am Computer mit der IP 192.168.10.10 ein/aus.

Wichtig ist, die Verbindung zu schliessen, da vMix das selber nicht macht.

```
Dim client =
WebRequest.Create("http://192.168.10.10:8088/api/?Function=OverlayInput1&Input=1")
Dim response = client.GetResponse()
response.Close
```

Diashow fertig, wechselt zu einem anderen Input, wenn alle Fotos durch sind

```
'Kopiere das alles in ein neues Skript, benenne das Skript und speiche es ab
'ersetze 13 durch die Inputnummer der Fotoliste
'ersetze 14 durch die Inputnummer auf die Du umschalten möchtest
'triggere deinen Fotoinput um das Skript auszuführen (ontransitionin -
scriptstart - Name des Skripts)
```

```
dim PhotoCompleted as string = ""
'eine kleine Verzögerung, um sicherzustellen, dass der Übergang beendet ist,
kann ebenfalls im Trigger eingestellt werden.
sleep(1000)
```

```
do while true
    dim xml as string = API.XML()
    dim x as new system.xml.xmldocument
    x.loadxml(xml)
```

```
    PhotoCompleted = (x.SelectSingleNode("//input[13]/@state").InnerText)
    if PhotoCompleted = "Completed"
        API.Function("Fade",Input:="14",Duration:="500") 'oder jede andere
        Funktion, die man ausführen möchte, nachdem alle Fotos angezeigt wurden
    exit do
```



```
end if

sleep(100)
loop
```

Externes Programm aus vMix starten

Startet ein externes Programm aus vMix und führt das Script weiter aus, wenn die externe Applikation geschlossen wird. Es kann z.B. ein komplexeres externes Programm ausgeführt werden, welches mit vMix Script nicht, oder nur schwer zu lösen wäre.

```
Dim P As New Process
P = Process.Start("notepad.exe")
P.WaitForExit()
```

Starten einer Anwendung (Beispiel Notepad mit angegebener vorhandener Datei)

```
Dim ExtApp As New ProcessStartInfo
ExtApp.FileName = "notepad.exe"
ExtApp.Arguments = "d:/demo.txt"
ExtApp.UseShellExecute = True
ExtApp.WindowStyle = ProcessWindowStyle.Normal
Dim proc As Process = Process.Start(ExtApp)
```

Variablenspeicher

Leider lässt vMix keine Public Variablen zu. Eine Variable ist immer nur im Script funktional und kann nicht von einem anderen Script verwendet werden. Da das Schreiben und Lesen in und aus einem Titel recht fix geht, kann man diesen als Variablenspeicher missbrauchen. Der Titel wird einfach an den Schluss des Projekts geladen und dann vergessen. Ein Beispieltitel hier

variablen.gtzip

Schreibt Variablen in den Titel, auch Integer immer als String in Gänsefüßchen! Muss ein Integerwert in einen String umgewandelt werden geschieht das mit cstr()

```
Input.Find("variablen.gtzip").Text("i1.Text")="111"
Input.Find("variablen.gtzip").Text("i2.Text")="222"
Input.Find("variablen.gtzip").Text("s1.Text")="irgend etwas bedeutendes"
```

Liest Variablen aus dem Titel. Beim Auslesen entsteht immer ein String, enthält dieser einen Zahlenwert, muss der String danach noch in einen Integerwert, cint(), gewandelt werden.

```
dim var1 as string = Input.Find("variablen.gtzip").Text("i1.Text")
dim var2 as string = Input.Find("variablen.gtzip").Text("i2.Text")
dim i1 as integer = cint(var1)
```

```
dim i2 as integer = cint(var2)
console.writeline(i1)
console.writeline(i2)
console.writeline(i1+i2)
```

File Funktionen

Directory erstellen, falls nicht vorhanden

```
dim Directory as string = "C:\Testdir"
If (Not System.IO.Directory.Exists(Directory)) Then
    System.IO.Directory.CreateDirectory(Directory)
End If
```

File erstellen, falls nicht vorhanden

```
dim Filename as string = "C:\Testdir\Datenbank.xml"
If (Not System.IO.File.Exists(Filename)) Then
    My.Computer.FileSystem.WriteAllText(Filename, " ", True)
End If
```

Files in einem Folder löschen

```
delete files in folder
Dim path As String = "C:\Testdir"

'löscht alle Files im Directory path
If Directory.Exists(path) Then

    For Each filepath As String In Directory.GetFiles(path)
        File.Delete(filepath)
    Next
    'löscht alle Subfolder im Directory path
    For Each d as string in Directory.GetDirectories(path)
        Directory.Delete(d, true)
    Next
End If
sleep(1000)
```

prüfen ob file vorhanden und dann ein Ereignis auslösen

```
dim FILELOCATION As string = "D:/"
dim IMAGENAME As string = ""
dim FILENAME As string = ""

IMAGENAME = "flagge.png"

FILENAME = FILELOCATION & IMAGENAME

If System.IO.File.Exists(FILENAME) Then
    'falls das file existiert, mach etwas
else
    'falls das file nicht existiert, mach etwas anderes
end if
```

alle Files aus einem Ordner in einen anderen kopieren

In diesem Beispiel werden die Files von dem Ordner E:\Irgendwo in einen Ordner RECORD auf dem Desktop kopiert.

```
Dim path As String = My.Computer.FileSystem.SpecialDirectories.Desktop +
"\RECORD"
If Directory.Exists(path) = False Then
    Try
        Directory.CreateDirectory(path)
    Catch ex As Exception
        'Fehlerbehandlung
    End Try
End If
My.Computer.FileSystem.CopyDirectory("E:\Irgendwo", path, True)
sleep(1000)
```

alle Files und Subfolder in einem Verzeichnis löschen

```
Dim path As String = "E:\TestDir"

'löscht alle Files im Directory path
If Directory.Exists(path) Then
    For Each filepath As String In Directory.GetFiles(path)
        File.Delete(filepath)
    Next
'löscht alle Subfolder im Directory path
    For Each d as string in Directory.GetDirectories(path)
        Directory.Delete(d, true)
    Next
End If
sleep(1000)
```

Zeit Funktionen

zu den letzten 10 Sekunden des aktiven Videos springen (beim Proben nützlich)

Für eine andere Zeit den Wert 9999 anpassen.

```
dim duration as string = ""
dim activeinput as string = ""

    dim xml as string = API.XML()
    dim x as new system.xml.xmldocument
    x.loadxml(xml)

activeinput = (x.SelectSingleNode("//active").InnerText)
duration = (x.SelectSingleNode("//input[@number='"& activeinput
&"']/@duration").Value)

API.Function("SetPosition",Input:=activeinput ,Value:=(duration-9999) )

sleep(500)
```

Wechsle Page in Titel (GT-Title) alle 5 Sekunden

Der Code ist für einen Titel, welcher 6 pages hat, welche nach 5 Sekunden automatisch gewechselt werden.

```
dim x as integer
Dim Index as integer =0
Do while Index <=100
for x = 1 to 6
sleep(5000)
api.Function("SelectIndex",Input:="Test.gtxml",Value:=x)
next x
index +=1
loop
```

Einen Titel zu einer bestimmten Tageszeit zusetzen

```
'löst eine Funktion zu einem bestimmten Tageszeitpunkt aus (16:00)
'in diesem Beispiel wird der Eingang 1 ("1") auf den Overlay1 gelegt

'hier die Zeit vorgeben
```

```
dim triggertime as string = "16:00"

Do While True
    If triggertime = DateTime.Now.ToString("HH:mm") Then      'bei 12H
Anzeige (AP/PM) hh:mm verwenden
        ' Hier kommt die Funktion, welche zu einem bestimmten Zeitpunkt
ausgelöst werden soll
        API.Function("OverlayInput1In",Input:="1")  '"ersetze "1" mit der
von dir gewünschten Input Nummer
        Exit do
    End If
    sleep(1000)
Loop
```

Streaming zu einem bestimmten Zeitpunkt starten

```
'löst eine Funktion zu einem bestimmten Tageszeitpunkt aus (16:00)
'in diesem Beispiel wird Streaming 1 gestartet
'Stream1:    Value:="0"
'Stream2:    Value:="1"
'Stream3:    Value:="2"

'hier die Zeit vorgeben
dim triggertime as string = "16:00"

Do While True
    If triggertime = DateTime.Now.ToString("HH:mm") Then
        ' Hier kommt die Funktion, welche zu einem bestimmten Zeitpunkt
ausgelöst werden soll, Start Streaming 1
        API.Function("StartStreaming",Value:="0")
        Exit do
    End If
    sleep(1000)
Loop
```

GT Titler Funktionen

growing bars (aufbauender Balken)



Die gtzip-Vorlage hat EINEN Balken, bestehend aus 579 einzelnen Textfeldern (harte Arbeit 😊, jeweils 3 Pixel breit, jeweils mit dem Zeichen „full block“ von Arial ■

[Lade den GT-Titel growingbars1.gtzip in vMix.](#)

download growingbar1.gtzip

Passe die Größe und Position des ersten Balkens mit den Positionsschiebern dieses Inputs an.
Erstelle 3 virtuelle Kopien von diesem Balken.
Verschiebe diese zusätzlichen Balken mit den Positions-Schiebereglern an die richtige Position in jedem Input.

(WICHTIG) Benenne die 3 Kopien um in:
growingbars2.gtzip
growingbars3.gtzip
growingbars4.gtzip

Lade einen transparenten Color-Input. Füge die 4 growingbar-Titel in den Multiviewer ein.
Dieser Eingang ist dann dein fertiges Overlay-Signal.

Damit sich die Balken bewegen, werden 2 Skripte benötigt.
eines, um die Balken zu löschen
eines, um die Balken wachsen zu lassen

Die Skripte werden mit dem Preset geladen.
Zum Testen lege bitte die files auf den Desktop.
-Projekt growingbars.vmix
-GT Titel growingbar1.gtzip

Im Skript growingbars können alle notwendigen Variablen geändert werden.
(Farbe, Transparenz des Balkens, Geschwindigkeit, Prozentsätze)
Das Skript ist für eine Summe von 100% ausgelegt.

Die Balkenbreite wird automatisch angepasst, d.h. der Teilnehmer mit der höchsten Prozentzahl erhält die volle Balkenbreite.

Tastenkürzel im Projekt:
C löscht die Balken
G startet den Balken

Alle benötigten Dateien (Projekt, Titel) sind hier

```
' this script is made for 4 bars, please extend it for more bars
dim i as integer          'variable for the loop
dim barcolor as string    'variable for the color of the bar
dim fieldname as string   'variable for the Fieldname of the Textfield
in the gtzip template
dim barwidth as integer = 579 '579 textfields, each 3 pixels wide
dim barspeed as integer = 0   'higher values = slower bar grow, 0 =
fastest possible
dim barpause as integer = 450 'waits xx miliseconfs, before painting the
next bar

'change here the values from your competition (sum of the 4 maximal 100%)
'dont change the part "dim Percentage_Candidate1 as integer =", only the
digits at the end of the line!
dim Percentage_Candidate1 as integer = 31
dim Percentage_Candidate2 as integer = 25
dim Percentage_Candidate3 as integer = 19
dim Percentage_Candidate4 as integer = 25

'this sets the color of the bars
'you can use all colors from here:
https://www.w3schools.com/colors/colors_names.asp
'color example "#FFD700" is GOLD
'normal color          = # + 6 HEX codes "#FFD700"
'transparent color    = # + 8 HEX codes, the first two characters are for
tranparence, "#CCFFD700" CC = 70% transparent

barcolor = "#CCFFD700"

' _____ nothing to change from here
_____

'this section calculates the maximal barwidth (500) to the winner percentage
dim Bar_width_1 as integer
dim Bar_width_2 as integer
dim Bar_width_3 as integer
dim Bar_width_4 as integer
dim maximalvalue as integer = Percentage_Candidate4
if Percentage_Candidate3 > Percentage_Candidate4 then maximalvalue =
Percentage_Candidate3
if Percentage_Candidate2 > Percentage_Candidate3 then maximalvalue =
Percentage_Candidate2
if Percentage_Candidate1 > Percentage_Candidate2 then maximalvalue =
Percentage_Candidate1
Bar_width_1 = barwidth /maximalvalue*Percentage_Candidate1
Bar_width_2 = barwidth /maximalvalue*Percentage_Candidate2
Bar_width_3 = barwidth /maximalvalue*Percentage_Candidate3
Bar_width_4 = barwidth /maximalvalue*Percentage_Candidate4
```

```
'this paints the bars, one after the other
for i = 1 to Bar_width_1
fieldname = "T"+cstr(i) + ".Text"
API.Function("SetTextColour",Input:="growingbar1.gtzip",SelectedName:=
fieldname ,Value:=barcolor)
SLEEP (barspeed)
next
SLEEP (barpause)

for i = 1 to Bar_width_2
fieldname = "T"+cstr(i) + ".Text"
API.Function("SetTextColour",Input:="growingbar2.gtzip",SelectedName:=
fieldname ,Value:=barcolor)
SLEEP (barspeed)
next
SLEEP (barpause)

for i = 1 to Bar_width_3
fieldname = "T"+cstr(i) + ".Text"
API.Function("SetTextColour",Input:="growingbar3.gtzip",SelectedName:=
fieldname ,Value:=barcolor)
SLEEP (barspeed)
next
SLEEP (barpause)

for i = 1 to Bar_width_4
fieldname = "T"+cstr(i) + ".Text"
API.Function("SetTextColour",Input:="growingbar4.gtzip",SelectedName:=
fieldname ,Value:=barcolor)
SLEEP (barspeed)
next
SLEEP (barpause)
```

Zufallszahl

[Zufallszahl per Script generieren. Titelbeispiel:](#)

wuerfel.gtzip

```
Static Generator As System.Random = New System.Random()
dim i as integer
dim fieldname as string = "Zahl.Text"
dim zahlrnd as integer

for i = 1 to 20
zahlrnd = Generator.Next(1, 6)
API.Function("SetText",Input:="wuerfel.gtzip",SelectedName:=
"Zahl.Text",Value:=zahlrnd )
sleep(100)
```


next

Zufallsfarben

Zufallsfarben in einem Titel, per Script generieren. Titelbeispiel:

textfarbe.gtzip

```
Static Generator As System.Random = New System.Random()  
dim i as integer  
dim fieldname as string ="Text.Text"  
dim color as string  
dim colorrnd as integer  
  
for i = 1 to 20  
colorrnd = Generator.Next(1, 11)  
'you can use colornames from  
https://www.w3schools.com/colors/colors\_names.asp  
if colorrnd = 1 then color = "aqua"  
if colorrnd = 2 then color = "blue"  
if colorrnd = 3 then color = "Chartreuse"  
if colorrnd = 4 then color = "Crimson"  
if colorrnd = 5 then color = "GreenYellow"  
if colorrnd = 6 then color = "orange"  
if colorrnd = 7 then color = "LawnGreen"  
if colorrnd = 8 then color = "red"  
if colorrnd = 9 then color = "yellow"  
if colorrnd = 10 then color = "white"  
API.Function("SetTextColour",Input:="textfarbe.gtzip",SelectedName:=  
fieldname ,Value:=color)  
sleep(100) 'damit man es sieht  
next
```

blinkender Text

SetTextVisible kann ein Textfeld ein-oder ausschalten (toggle)

SetTextVisibleON kann ein Textfeld einschalten

SetTextVisibleOFF kann ein Textfeld ausschalten

das selbe kann mit einem Bild (Image) gemacht werden:

API.Function(„SetImageVisible“,Input:=„visibledemo.gtzip“,SelectedName:=„Image1.Source“)

Die im GT Titler erzeugten Rechtecke oder sonstigen Formen können leider weder umgefärbt noch sichtbar/unsichtbar gemacht werden.

Beispieltitel hier:

visibledemo.gtzip

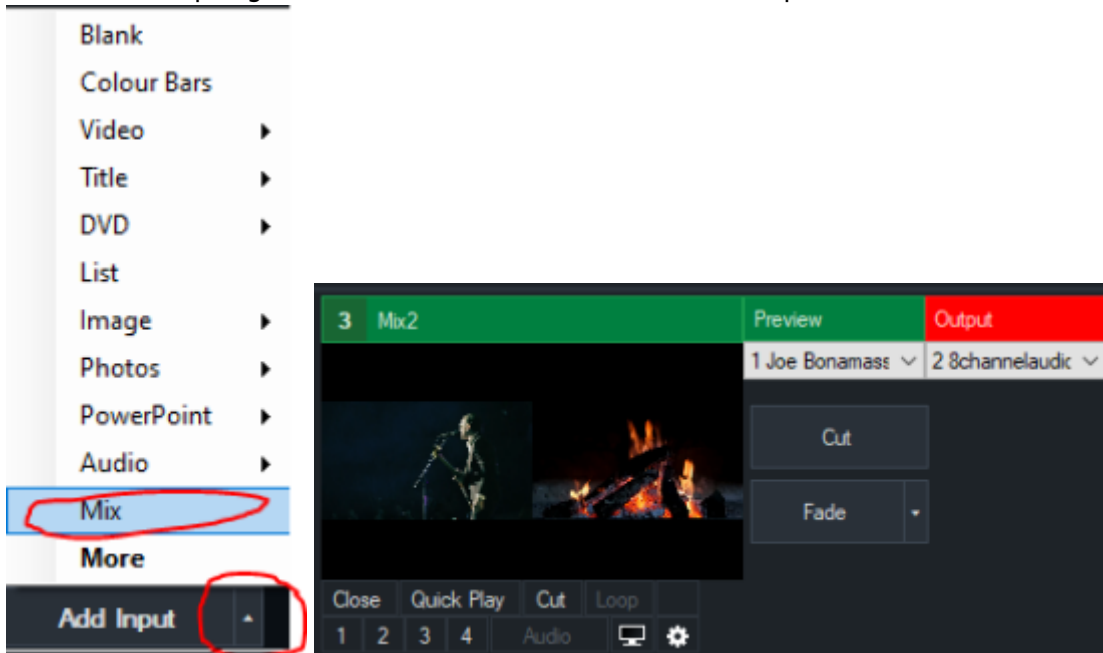
```
do while true
API.Function("SetTextVisible",Input:="visibledemo.gtzip",SelectedName:="Text
Block1.Text")
API.Function("SetImageVisible",Input:="visibledemo.gtzip",SelectedName:="Ima
gel.Source")
sleep(500)
loop
```

MIX Funktionen

Mixer Input

vMix hat noch 3 kleine Mixer Inputs, welche auch gut für verschiedene scriptgesteuerte Funktionen verwendet werden können. Der Vorteil der 3 zusätzlichen Mischer ist, dass jeder Eingang auch geblendet werden kann. (Die Mischer sind nur ab der 4K Version verfügbar)

Einen Mixer Input generiert man mit dem kleinen Aufwärtspfeil neben dem Button Add Input.



```
' setze Input 1 in Mix2 Preview
API.Function("PreviewInput",Input:=1,Mix:=1)

' setze Input 1 in Mix2 Output
API.Function("ActiveInput",Input:=1,Mix:=1)

' überblende in Mix2
API.Function("Fade",Input:=0,Mix:=1)      ' INPUT 0 ist immer Programm
```

XPATH finden

Es ist manchmal etwas verwirrend/knifflig, den richtigen xpath zu finden, um bestimmte Daten aus einer XML-Datei abzurufen, sei es innerhalb eines Skripts oder im XML-Datenquellenmanager. Ein paar hilfreiche Links, die einen xpath für dich generieren können. Allerdings muss man die Ergebnisse möglicherweise etwas anpassen, je nachdem, ob man sie in einem Skript oder in einem Datenquellen-Manager verwendet.

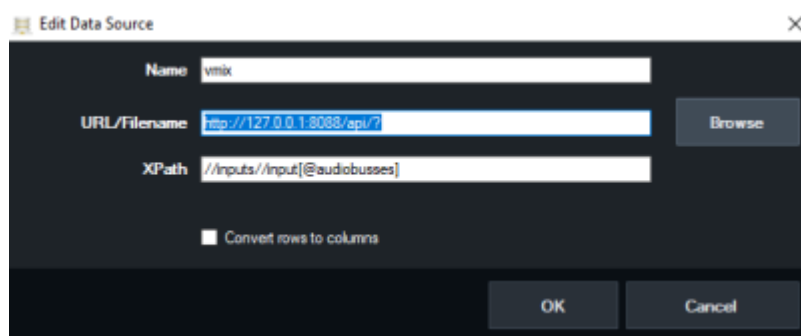
http://xmltoolbox.appspot.com/xpath_generator.html (liefert ein Ergebnis basierend auf einer Auswahl)

<https://www.easycodewall.com/XPathUtility.jsp> (gibt eine ganze Liste möglicher xpaths aus, hat auch noch ein paar andere Goodies)

Datasource

Mit Datasource können Textfelder eines Titels mit externen Daten gekoppelt werden. Als Daten funktionieren:

-Excel -Google Sheet -JSON -RSS -Text -XML

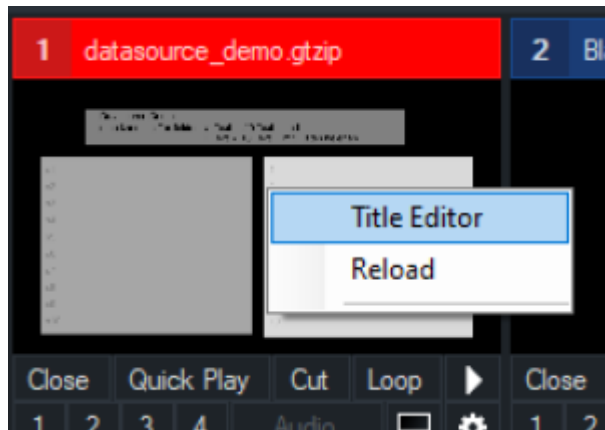


API von vMix als Datasource nutzen

Da vMix seine API Daten als XML zur Verfügung stellt, kann der Datasource Manager von vMix natürlich auch seine eigenen Daten lesen und auswerten. Dazu öffnet man einen Titel-Input und clickt mit der rechten Maustaste in das kleine Titelfeld. Demotitel hier:

datasource_demo.gtzp

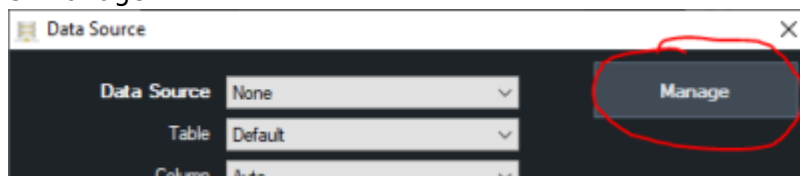
1. Rechte Maustaste ins Titelfenster, Title Editor



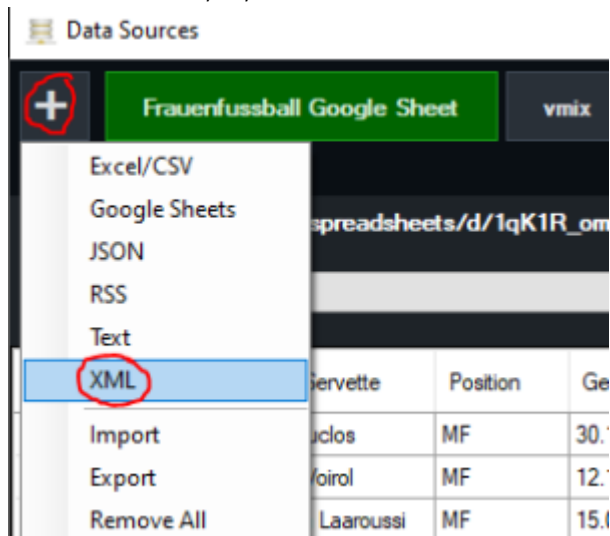
2. Data Source



3. Manage



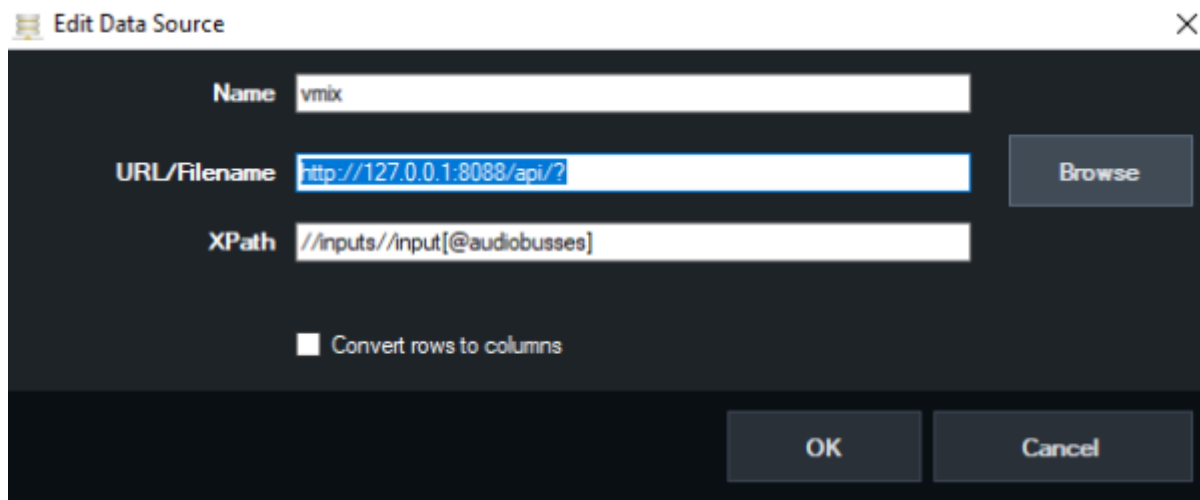
4. Data Source, +, XML



5. Name,

z.B. vmix API, <http://127.0.0.1:8088/api/?>, und einen gültigen XML node wie z.B. `//inputs//input[@audiobusses]`

Dann OK drücken.



Edit Data Source

Name: vmix

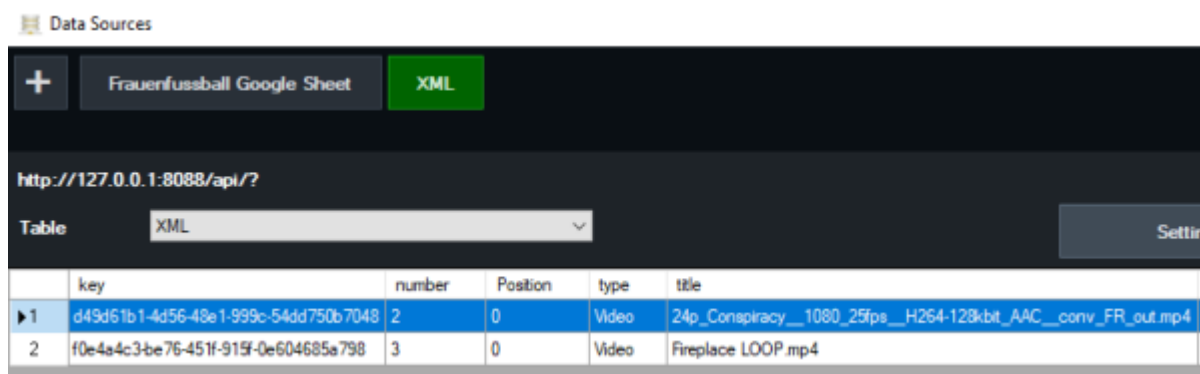
URL/Filename: `http://127.0.0.1:8088/api/?` Browse

XPath: `//inputs//input[@audiobusses]`

☐ Convert rows to columns

OK Cancel

6. Das sollte dann so aussehen:



Data Sources

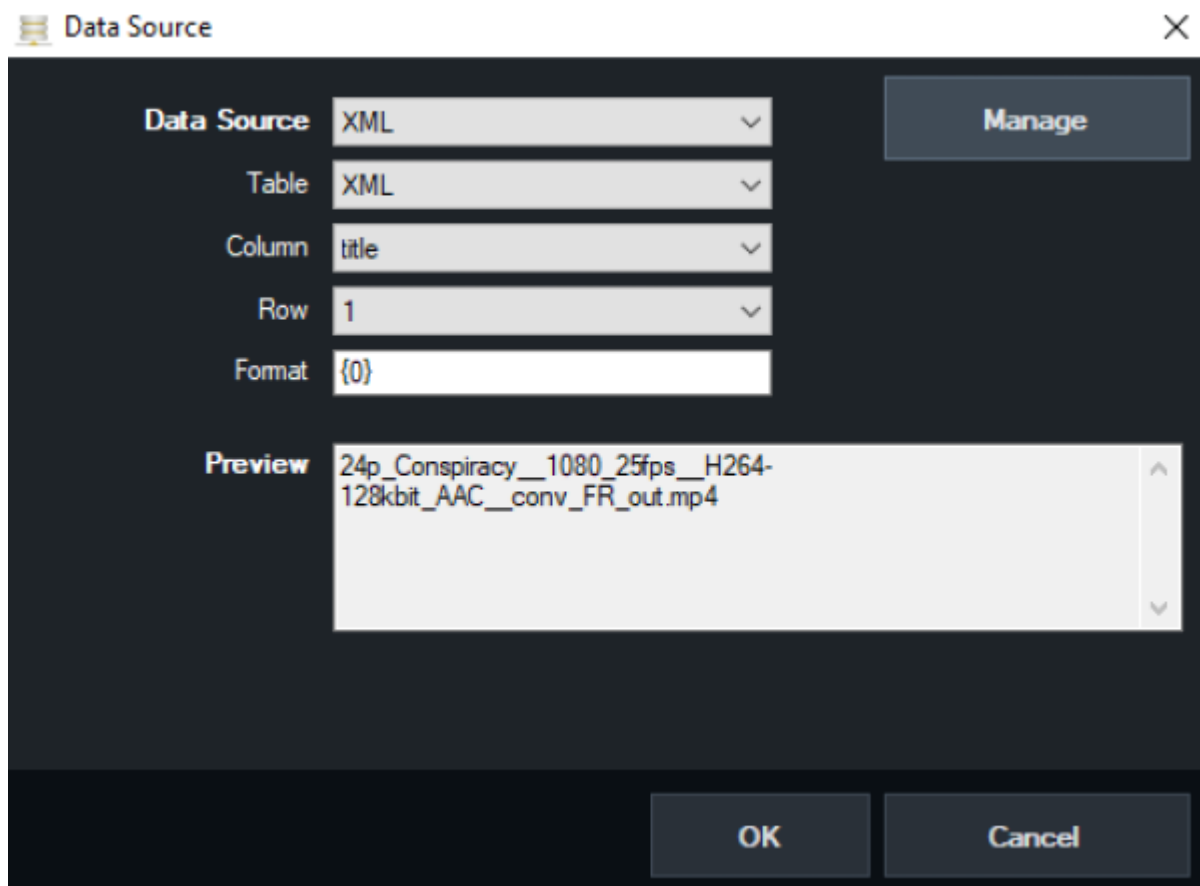
+ Frauenfussball Google Sheet XML

http://127.0.0.1:8088/api/?

Table: XML

	key	number	Position	type	title
1	d49d61b1-4d56-48e1-999c-54dd750b7048	2	0	Video	24p_Conspiracy__1080_25fps__H264-128kbit_AAC__conv_FR_out.mp4
2	f0e4a4c3-be76-451f-915f-0e604685a798	3	0	Video	Fireplace LOOP.mp4

7. Danach können die einzelnen Daten irgendwelchen Textfeldern im Titel zugewiesen werden.



Data Source

Data Source: XML Manage

Table: XML

Column: title

Row: 1

Format: {0}

Preview: 24p_Conspiracy__1080_25fps__H264-128kbit_AAC__conv_FR_out.mp4

OK Cancel

Beachte: es kommen nur Daten, wenn auch etwas ins XML geschrieben wird. Falls keiner der Inputs über Audio verfügt, kommt bei dem Node vom Beispiel einfach NICHTS.

eine bestimmte Datenzeile per script wählen (Excel oder Google sheet)

Im Beispiel heisst das Google Sheet „Frauenfussball Google Sheet“ und die Tabelle „Servette“. Wenn man nun Titeldaten mit selected row verknüpft, kann per API eine Zeile der Tabelle gewählt werden. So kann eine Tabelle durchgeblättert werden.

```
API.Function("DataSourceSelectRow",Value:="Frauenfussball Google Sheet,Servette,2")
```

"rotating" Datasource

zeigt, bis das script von Hand gestoppt wird, alle 5 Sekunden eine von 10 Zeilen aus einer Tabelle Google Sheet „Frauenfussball Google Sheet“ und die Tabelle „Resultate“.

```
while (true)
For i As Integer = 1 to 10
API.Function("DataSourceSelectRow",Value:="Frauenfussball Google Sheet,Resultate," &i)
    Sleep(5000)
Next
end while
```

Streaming

streamingkey/URL/PW setzen

RTMP Stream URL für Stream1
(Stream2 „Input:=2“, Stream 3 „Input:=3“)

```
API.Function("StreamingSetURL","Input=1",
"rtmp://x.xx123456.i.akamaientrypoint.net/EntryPoint")
```

Streamkey

```
API.Function("StreamingSetKey","Input:=1", "2222oioioi23456")
```

OPTIONAL

User

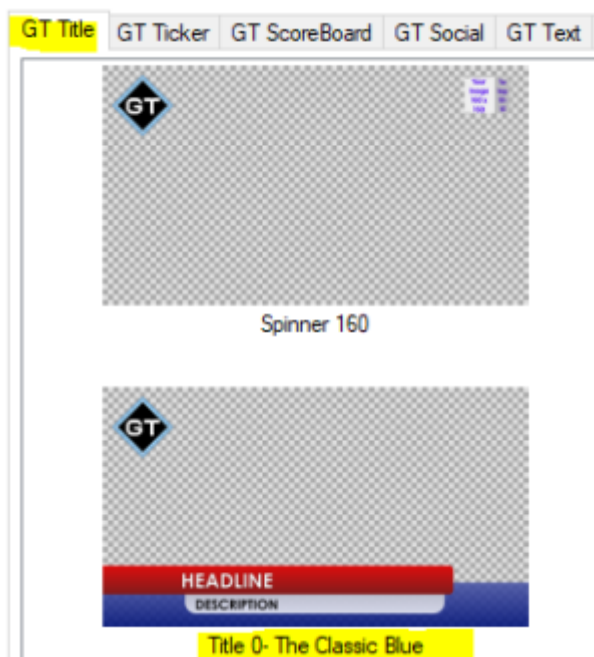
```
API.Function("StreamingSetUsername","Input:=1", "USER99")
```

Passwort

```
API.Function("StreamingSetPassword","Input:=1", "123456")
```

läuft das streaming (stream1)?

Streamingstatus via API abfragen und als Anzeige den Wert in einem Title Input schreiben. Als Titel ist ein Beispieltitel aus der GT Library gewählt. **Title 0- The Classic Blue.gtzip**



```
dim isstreaming as string = ""

' do while true/loop prüft stetig, bis das script angehalten wird
do while true

    dim xml as string = API.XML()
    dim x as new system.xml.xmldocument
    x.loadxml(xml)

'sind wir am streamen?
    isstreaming = (x.SelectSingleNode("/vmix/streaming[1]").InnerText)
'eine Antwort in einen Titel schreiben und die Farbe entsprechend ändern
if isstreaming = true
    API.Function("SetText",Input:="Title 0- The Classic
Blue.gtzip",SelectedName:="Headline.Text",Value:="Streaming LÄUFT")
    API.Function("SetTextColour",Input:="Title 0- The Classic
Blue.gtzip",SelectedName:="Headline.Text",Value:="red")
```

```
else
  API.Function("SetText",Input:="Title 0- The Classic
Blue.gtzip",SelectedName:="Headline.Text",Value:="Streaming STOP")
  API.Function("SetTextColour",Input:="Title 0- The Classic
Blue.gtzip",SelectedName:="Headline.Text",Value:="green")
end if

sleep(500)
loop
```

PTZ Kameras bewegen

zoomen

```
API.Function("SetZoom",Input:="1",Value:="0.5")
```

bewegen

```
API.Function("SetPanX",Input:="1",Value:="0.5")
```

```
API.Function("SetPanY",Input:="1",Value:="0.5")
```

From:
<https://tvcrew.ch/vmix/> - **vMix Wiki Deutsch**

Permanent link:
<https://tvcrew.ch/vmix/doku.php?id=scripting&rev=1617611611>

Last update: **2023/10/20 20:26**

