

Scripting

Scripting funktioniert nur bei der 4K oder PRO Version.

Einige Beispiele und Tricks für die Scripts in diesem Beitrag 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 Web scripting und VB.NET beim Schreiben deines Skripts. Alle Variablen sind Lokal, Ab Version 24 gibt es noch dynamische Variablen und dynamische Inputs, welche man als public Variablen bezeichnen könnte. Deren Anwendung ist aber etwas kryptisch.. Man kann aber Textfelder eines nicht benutzten Titel Eingangs sehr einfach als Public Variablen zweckentfremden.

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

THE UNOFFICIAL vMix API Reference

Das ultimative Super Onlinetool, Zeigt zu jedem möglichen Befehl die korrekte Syntax, um vMix zu steuern. Aktuell mit code bis vMix26.

[THE UNOFFICIAL vMix API Reference](#) Link mit Erlaubns vom Autor Nick Roberts.

Syntax zu:

- Companion custom command (im Companion Browser | Studiocoast:vMix PRESETS → General)
- Web scripting
- VB.NET scripting
- HTTP GET request
- TCP packet ASCII

vMix Script Builder

Am Anfang ist der vMix Script Builder (ein externes Hilfsprogramm) auch eine unverzichtbare Hilfe. Es erstellt Scripts mit der korrekten Syntax, Auswahl via Pulldown Menus.

[vMix Script Builder](#) Der [Universal Title Controller](#) stammt vom gleichen Autor. Link mit Erlaubnis vom Autor Richard „Raugert“

vMix API

Onlinetool, zeigt die korrekte Syntax, um vMix via HTTP commands zu steuern.

[online HTTP vMix commands Builder](#) Link mit Erlaubns vom Autor Jeff „Dist“ Martin. <https://dist.dev/>

vMix Script Editor

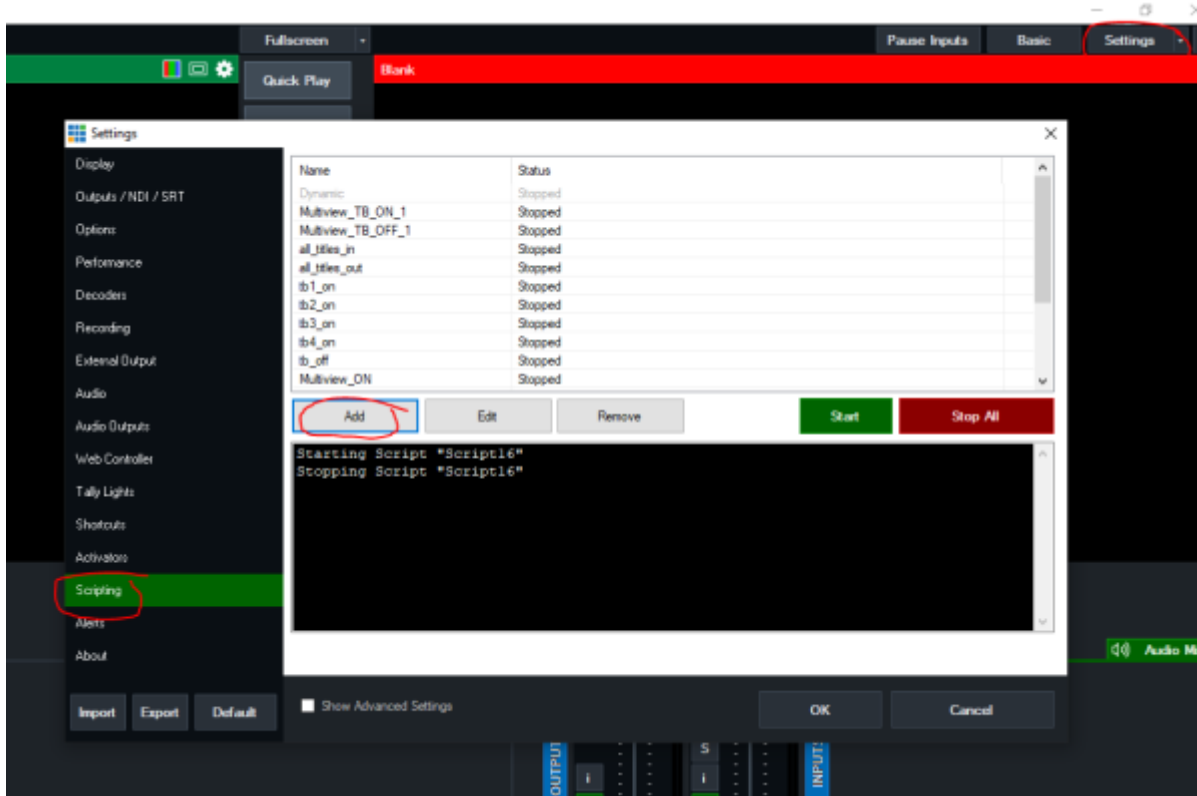
Die Beschreibung auf der vMix Seite ist etwas dürrtig, trotzdem lohnt sich immer ein Blick dorthin.

[vMix API Website](#)

Eine Liste der vorhandenen API Funktionen gibt es auf der Webseite von vMix.

[alle API Funktionen Version 27 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
Script Name

Title: []

Description 1: []

Description 2: []

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 audiobusoff.

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.

WEB Scripting

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

VB.NET API

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

HTTP GET request

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

VB.NET Input.Find Input.Find ist ein vMix Sonderkommando

```
Input.Find("Title 0- The Classic Blue.gtzip").Text("Headline.Text")= "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 „kodierte“ 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%2C  
Sheet1%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!"
```

Inhalt eines Textfeldes aus einem Titel auslesen und in eine Variable speichern

Man kann einen Inhalt eines Textfeldes auslesen, um z.B. zu prüfen, ob ein Wert erreicht ist, oder anhand des Textes irgend eine Funktion per Script auszulösen.

```
dim läuftts as string  
  
läuftts = Input.Find("multiviewer_rotlicht.gtzip").Text("run.Text")
```

Das nächste Beispiel macht genau das. Im Titel ist ein Textfeld, welches run.Text heisst. Das Feld ist im Editor auf nicht sichtbar gestellt. D.h. es ist im Titel online unsichtbar, kann aber als variables Textfeld benutzt werden. In diesem Fall ist der Text von run.Text entweder „0“ oder „1“.



```
dim läufsts as string

läufsts = Input.Find("multiviewer_rotlicht.gtzip").Text("run.Text")

if läufsts = "0" then
API.Function("StartCountdown", Input:="multiviewer_rotlicht.gtzip",
Selectedname:="total.Text")
API.Function("SetText",Input:="multiviewer_rotlicht.gtzip",SelectedName:="run.
n.Text",Value:="1")
API.Function("StartCountdown", Input:="multiviewer_rotlicht.gtzip",
Selectedname:="sequenz.Text")
end if

if läufsts = "1" then
API.Function("StopCountdown", Input:="multiviewer_rotlicht.gtzip",
Selectedname:="sequenz.Text")
API.Function("SetCountdown", Input:="multiviewer_rotlicht.gtzip",
Value:="09:00:00", Selectedname:="sequenz.Text")
API.Function("SetText", Input:="multiviewer_rotlicht.gtzip",
Value:="00:00:00", SelectedName:="sequenz.Text")
API.Function("StartCountdown", Input:="multiviewer_rotlicht.gtzip",
Selectedname:="sequenz.Text")
end if
```

Output2 (ändern)

Eingang 1 Video auf **OUTPUT2** routen

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

Input

Es gibt 3 Möglichkeiten, in Scripting einen bestimmten Eingang in vMix zu verwenden:

- Über die Eingangsnummer: 1 bis xx
- GUID (auslesen via XML): 6a496926-9aea-462a-b104-944a5f928602
- Nach Name: Bild.jpg

Verwendest Du Nummern als Eingabe, denke daran, dass die Nummer eines Inputs ändern kann, wenn du Inputs löscht oder hinzufügst

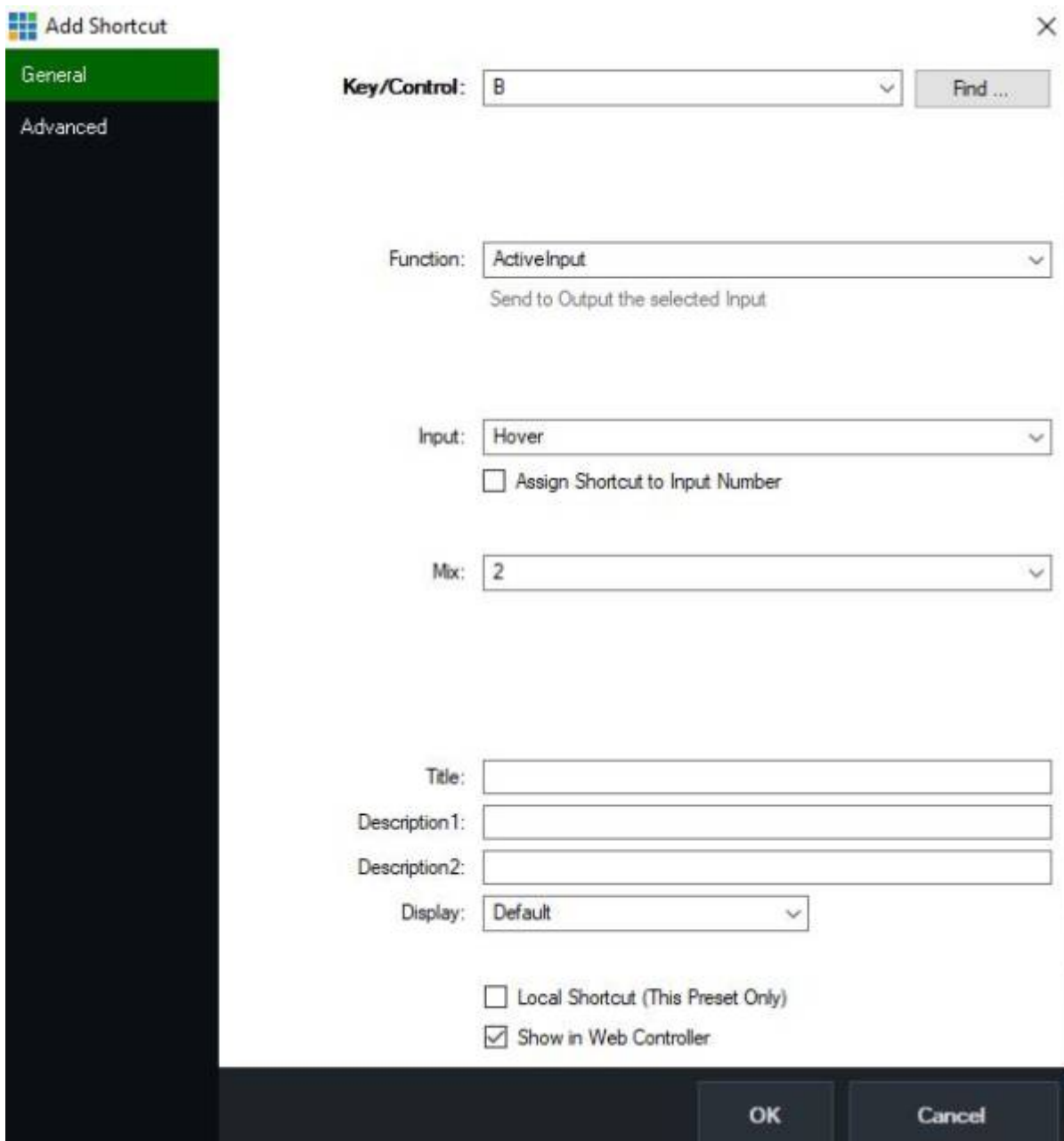
Spezielle Input Nummern:

- 0 ist Preview
- -1 ist PGM
- -3 Eingabe, die sich derzeit unter dem Mauszeiger befindet (Hover)

Input Hover (bezieht sich auf den vorherigen INPUT Text)

In Scripting macht -3 keinen Sinn, da der Fokus der Maus meistens dann nicht über einem Input sein wird. In Shortcuts aber sehr wohl! Der nächste Shortcut schaltet den Eingang auf welchem sich der Mauszeiger befindet, auf Mix 1 (in Shortcuts Mix2, da Mix1 der Main Output ist).

Bei einer Gewinnershow könnte man so den Gewinner markieren und der Regisseur kann dann bloss noch auf den SIEGER Shortcut drücken.



welcher Input ist aktiv am OUTPUT

(mix1 in den shortcuts oder mix0 in den scripts)

```
do while true
  ' XML-Daten von vMix abrufen und in der Variable xml speichern
  dim xml as string = API.XML()
  ' Ein neues XML-Dokumentobjekt erstellen und die XML-Daten darin laden
  dim x as new system.xml.xmldocument
  x.loadxml(xml)
  ' Den Wert des <active>-Knotens unter <vmix> extrahieren und als
  Zeichenfolge in der Variablen activenumber speichern
  dim activenumber as string =
  (x.SelectSingleNode("//vmix/active").InnerText)
  ' activenumber enthält nun den Wert (Zahl) des gerade aud PGM aktiven
```

```

Eingangs als string
  ' Den Text eines Eingangs namens "Title 0- The Classic Blue.gtzip" auf
den Wert des gerade aktiven Inputs setzen, der in der Variablen activenumber
gespeichert ist
  API.Function("SetText", Input:="Title 0- The Classic Blue.gtzip",
SelectedName:="Headline.Text", Value:="" + activenumber)
  ' Die Ausführung des Skripts für 1000 Millisekunden (1 Sekunde)
pausieren
  sleep(1000)
loop

```

welcher Input ist aktiv am PREVIEW

(mix1 in den shortcuts oder mix0 in den scripts)

Dies ist der gleiche Code wie im vorherigen Beispiel, nur wird anstelle des PGM Wertes der PREVIEW Wert gelesen.

```

do while true
  dim xml as string = API.XML()
  dim x as new system.xml.xmldocument
  x.loadxml(xml)
  dim activenumber as string =
(x.SelectSingleNode("//vmix/preview").InnerText)
  API.Function("SetText", Input:="Title 0- The Classic Blue.gtzip",
SelectedName:="Headline.Text", Value:="" + activenumber)
  sleep(1000)
loop

```

Script um den aktiven Input auf den nächsten zu schalten

Schaltet Output auf 2 wenn der Output auf 1 ist, auf 3 wenn er auf 2 ist etc.

```

'get the API data from vMix
Dim xml As String = API.XML()
Dim x As New System.Xml.XmlDocument()
x.LoadXml(xml)

'get active input
Dim activeNumber As Integer =
Integer.Parse(x.SelectSingleNode("/vmix/active").InnerText)

' get number of inputs
Dim inputCount As Integer = x.SelectNodes("/vmix/inputs/input").Count

'increase by one
Dim nextInput As Integer = activeNumber + 1

```

```
'this is, if you want to do something at the beginning of the inputs (1), it
starts again at the last one.
If nextInput > inputCount Then nextInput = 1 ' wrap

' cut to the next input
API.Function("cutdirect",nextInput)
```

Script um den aktiven Input auf den vorhergehenden zu schalten

Schaltet Output auf 4 wenn der Output auf 5 ist, auf 3 wenn er auf 4 ist etc.

```
'get the API data from vMix
Dim xml As String = API.XML()
Dim x As New System.Xml.XmlDocument()
x.LoadXml(xml)

'get active input
Dim activeNumber As Integer =
Integer.Parse(x.SelectSingleNode("/vmix/active").InnerText)

'get number of inputs
Dim inputCount As Integer = x.SelectNodes("/vmix/inputs/input").Count

'decrease by one
Dim prevInput As Integer = activeNumber - 1

'wrap , returns to the highest if at 1
If prevInput < 1 Then prevInput = inputCount

'cut to the previous input
API.Function("cutdirect", prevInput)
```

Input auf einen der Outputs schneiden

Ab Version 25 besitzt vMix bis zu 16 sogenannte Mix Inputs (Aux oder M/E) mehr als nur den Output (Mix1 oder Mix0 in der API) Diese können dann auf andere Outputs geroutet werden. (max.6 Outputs in vMix) Also auch Mix setzen! Ab Version 27 können mehr als zwei EXTERNAL OUTPUTS gesetzt werden, dies hängt von der Performance des Rechners ab. Ab einer QUADRO RTX 4000 funktionieren 4 einwandfrei. Ab den GeForce RTX Modellen der 3000er Serie funktionieren auch mehr. Wenn der Rechner aber noch Slow macht oder massiv aufzeichnet, dann würde ich auf diese Funktionen nur mit einer neuen A Serie Karte zugreifen.

Eingang 4 Video auf OUTPUT routen

```
Function=Activeinput&Input=2&Mix=0
oder
```

```
http://127.0.0.1:8088/API/?Function=Cut&Input=4&Mix=0
```

Eingang 4 auf OUTPUT (Mix0) direkt schneiden mit Cutdirect (Cutdirect geht nur auf MIX0/PGM)

```
Function=Cutdirect&Input=4  
oder  
API.Function("Cutdirect",4,)
```

Eingang 4 Video auf Mix1 Input (AUX1) routen

```
Function=Activeinput&Input=2&Mix=1  
oder  
http://127.0.0.1:8088/API/?Function=Cut&Input=4&Mix=1
```

Eingang 3 Video auf Preview routen

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

Eingang mit einem bestimmten Namen auf die Linie routen

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

Hartschnitt von Eingang2 auf Mix1

```
http://192.168.10.102:8088/api/?Function=Activeinput&Input=2&Mix=1
```

Schneidet Input 4 zu PGM (Cut)

Cut schneidet den Input 4 auf den Ausgang und wechselt den vorherigen Input auf Preview, Cutdirect 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 oder Mix

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)
```

fade input 1, 500ms fade time to PGM (mix0)

```
Function=Fade&Input=1&Duration=500&Mix=0
```

fade SDI Input, 300ms fade time to Mix Input1 (mix1)

```
API.Function("Fade", Input:="DeckLink Duo (1) 1", Duration:="300", 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
```

g

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
```

Video

Holt das zuletzt gespeicherte Video aus einem Verzeichnis

```
Dim szExt As String = ".mp4"
    Dim szFolder As String =
System.Environment.GetFolderPath(System.Environment.SpecialFolder.MyDocument
s) & "\Videos"
    Dim szLatestFile As String = ""
    Dim dLatestFile As Date = Nothing
    For Each szFile As String In System.IO.Directory.GetFiles(szFolder)
        Dim fi As New System.IO.FileInfo(szFile)
        If fi.Extension = szExt Then
            If fi.LastWriteTime > dLatestFile Then
                szLatestFile = szFile
                dLatestFile = fi.LastWriteTime
            End If
        End If
    Next
    If System.IO.File.Exists(szLatestFile) Then
        API.Function("AddInput", Value:="Video|" & szLatestFile)
    End if
```

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:=Timeremaining)
    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

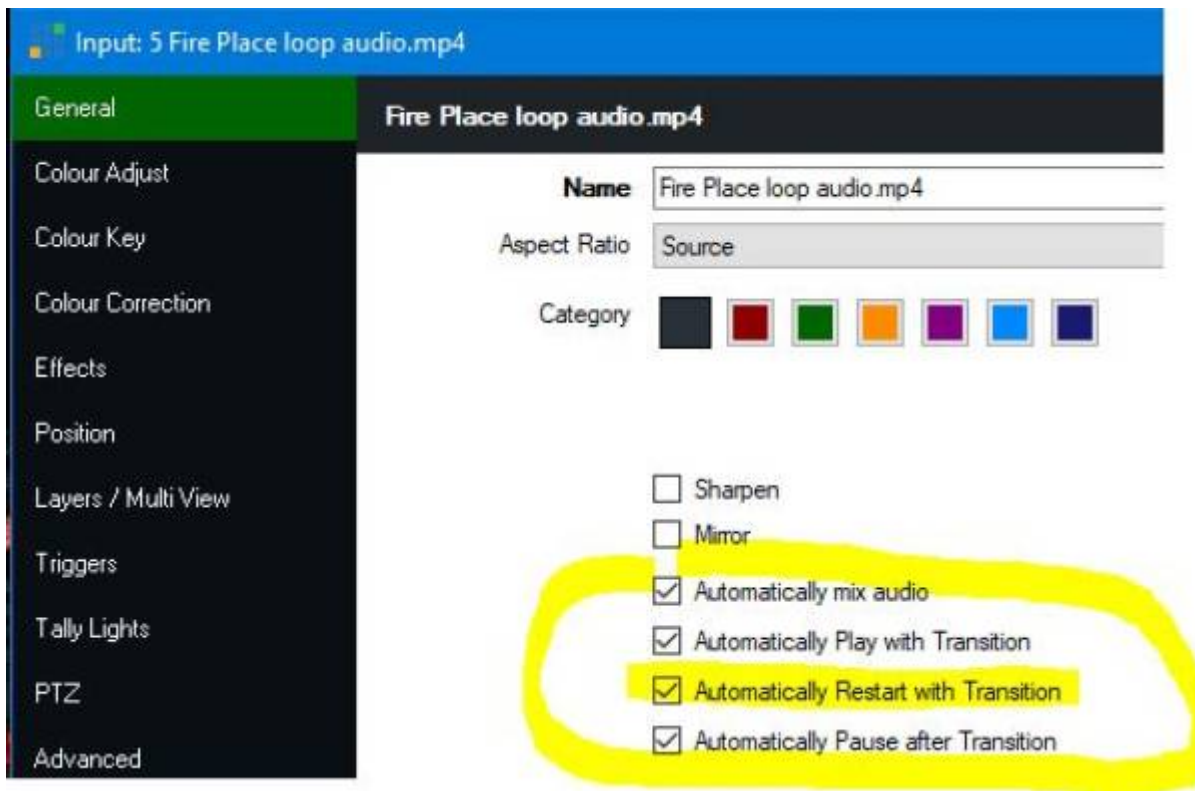
    sleep(500)
Loop
```

Video auf eine bestimmte Stelle positionieren

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

Play Video in Input 5

Beim starten eines Video Inputs sind die General-Settings des Inputs wichtg. Soll das Video jedesmal von Anfang an gestartet werden, pausieren etc.



Dies startet nur das Video auf Input 5, **schneidet es aber nicht irgenwohin**

```
Function=Play&Input=5
```

Falls Das Video auch gleichzeitig beim Starten auf den Ausgang geschnitten werden soll, dann muss nach dem Startbefehl auch noch der Cut Befehl mitgeschickt werden. Hier ist wieder Mix=0 der OUTPUT/PGM, Mix=1 der erste Input-MIX, welcher dann aber im kleinen Vorschauenfenster mit Mix2 angeschrieben ist.... Verwirrend, ist aber so.

```
Function=Play&Input=5
Function=Activeinput&Input=5&Mix=0
```

Audio

Audio Bus der Quelle 1 Einschalten (grüner Lausprecher)

```
Function=Audio0N&Input=1
oder
API.Function("Audio0N",1,)
```

Audio Bus der Quelle 1 auf Audio Ausgang D routen

```
Function=AudioBus0N&Input=1&Value=D
oder
API.Function("AudioBus0ff",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
```

schaltet alle Audio Ausgänge um (toggle)

```
'Toggles on/off all Audio Busses  
Dim busValues As String() = {"A", "B", "C", "D", "E", "F", "G"}  
For Each value As String In busValues
```

```
API.Function("BusXAudio", Value:=value)
Next
```

schaltet alle audio ausgänge AUS

```
Dim busValues As String() = {"A", "B", "C", "D", "E", "F", "G"}
For Each value As String In busValues
    API.Function("BusXAudioOff", Value:=value)
Next
```

schaltet alle audio ausgänge EIN

```
Dim busValues As String() = {"A", "B", "C", "D", "E", "F", "G"}
For Each value As String In busValues
    API.Function("BusXAudioOn", Value:=value)
Next
```

schaltet Audio Master

```
API.Function("MasterAudioOff")
```

```
API.Function("MasterAudioOn")
```

oder toggle

```
API.Function("MasterAudio")
```

Schaltet bei allen Audio Ausgängen Solo Aus

```
'switch off all solos on busses
Dim busValues As String() = {"A", "B", "C", "D", "E", "F", "G"}
For Each value As String In busValues
    API.Function("BusXSoloOff", Value:=value)
Next
```

```
'switch off all solos on inputs
Dim xmlDoc As New XmlDocument()
Dim i as integer
```

```
xmlDoc.LoadXml(API.Xml)
Dim inputNodes As XmlNodeList = xmlDoc.SelectNodes("/vmix/inputs/input")
Dim inputCount As Integer = inputNodes.Count
for i = 1 to inputcount
    API.Function("SoloOff", i)
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)

'gets active title from the playlist
dim word as string =
(x.SelectSingleNode("//input[@number=2]/@title").InnerText)

'removes .mp3 from the title
word = word.remove(word.Length - 4)

' Split the string by "-"
Dim parts() As String = word.Split(New Char() {"-"},
StringSplitOptions.RemoveEmptyEntries)

' Trim each part to remove leading and trailing spaces
For i As Integer = 0 To parts.Length - 1
    parts(i) = parts(i).Trim()
Next

'gets number of parts
```

```
Dim number As Integer = parts.Length - 1

'shows each part of the tile for a second
for ii as integer = 1 to number
API.Function("SetText",Input:="Title 33- On the shelf-
Peach.gtzip",SelectedName:="Headline.Text",Value:=parts(ii))
sleep(1000)
next
loop
```

konvertieren eines API XML Wertes in einen Lautstärkewert 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
```

Sidechain Ducker Script

Dieses Script „duckt“ ein Original Audio Signal wenn z.B. der Uebersetzer spricht.
Das Script stammt von einem Russischen Prorammiierer, es gibt ein Video auf YouTube dazu.
<https://youtu.be/amrkD2SldrM>

```
' ---- settings for SIDECHAIN control script ----
dim translateInput as string = "Mice" 'Name of the vMix Input with the
Translator
dim origInput as string = "Orig PGM" 'Name of vMix Input with the original
soundtrack

dim volumeStandart as string = "100" 'The volume of the Original Reference
when the Translator is silent
dim fadeTimeStandart as string = "1200" 'The length of the volume increase
when the Interpreter is silent

dim volumeSpeaking as string = "60" 'The volume of the Original when the
Translator speaks
dim fadeTimeSpeaking as string = "200" 'The length of time Original is
silenced when the Interpreter speaks

dim voicethreshold as string = "0.05" 'Interpreter response threshold, from
0 to 1 (log scale)
' Corresponds to the values (not the length) on the volume slider:
0.5=50%~-6dB, 0.1=10%~-20dB...
' The response may not work with a single signal pulse, but works well with
a signal like voice or music

dim checkingIter as integer = 10 'The number of iterations of checking the
interpreter silence before triggering
dim checkingIterTime as integer = 200 ' The interval (milliseconds) between
the checking iterations (recommned from 100, silent 200)
' Means the length of the check that the translation is silent and can be
turned on Orig. = checkingIter * checkingIterTime (millisecs)
```

```
'---- end of settings----

dim voicemeter as string = ""
dim speaking as boolean = false
dim silence as boolean = false
dim silencechecking as integer = 0
dim muted as boolean = false

do while true

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

voicemeter = (x.SelectSingleNode("//input[@title='"& translateInput
&''']/@meterF1").Value)
if (x.SelectSingleNode("//input[@title='"& translateInput
&''']/@meterF2").Value) > voicemeter
voicemeter = (x.SelectSingleNode("//input[@title='"& translateInput
&''']/@meterF2").Value)
end if
muted = (x.SelectSingleNode("//input[@title='"& translateInput
&''']/@muted").Value)

'API.Function("SetText",Input:="TextInput",SelectedName:="Timer.Text"
,Value:=voicemeter)
'console.writeline(voicemeter)

if voicemeter > voicethreshold And voicemeter.IndexOfAny("[E]".ToCharArray)
= -1 And Not muted

  if silencechecking >= checkingIter '<> 0 '!=

    'console.writeline("Speaking")
    silencechecking = 0
'API.Function("SetTextColour",Input:="TextInput",SelectedName:="Timer.Text",
Value:="yellow")
'API.Function("SetText",Input:="TextInput",SelectedName:="Timer.Text"
,Value:="Translation speaks! " + voicemeter)
    Input.Find(origInput).Function("SetVolumeFade", volumeSpeaking + ","
+ fadeTimeSpeaking)

  end if

  'console.writeline(silencechecking)

else

  if silencechecking < checkingIter
    silencechecking += 1
```

```

    end if
    if silencechecking = checkingIter
        'console.writeline("Silence")
'API.Function("SetTextColour",Input:="TextInput",SelectedName:="Timer.Text",
Value:="white")
'API.Function("SetText",Input:="TextInput",SelectedName:="Timer.Text"
,Value:="Translation is silent! " + voicemeter)
        Input.Find(origInput).Function("SetVolumeFade", volumeStandart + ","
+ fadeTimeStandart)
        silencechecking = checkingIter + 1
    end if

    'console.writeline(silencechecking)
end if

sleep(checkingIterTime)

Loop

```

Sidechain Ducker Script DEUTSCHE ERKLÄRUNG

Dieses Script „duckt“ ein Original Audio Signal wenn z.B. der Uebersetzer spricht.
Das Script stammt von einem Russischen Prorammierer, es gibt ein Video auf YouTube dazu.
<https://youtu.be/amrkD2SldrM>

```

' ---- Einstellungen für das SIDECHAIN control script ----
dim translateInput as string = "Mik" 'Name des vMix-Inputs mit dem
Übersetzer Mik
dim origInput as string = "Orig PGM" 'Name des vMix-Inputs mit der Original-
Tonspur (Original)

dim volumeStandart as string = "100" 'Die Lautstärke des Originals, wenn der
Übersetzer schweigt
dim fadeTimeStandart as string = "1200" 'Die Zeit mit der die Lautstärke des
Originals zunimmt, wenn der Dolmetscher schweigt

dim volumeSpeaking as string = "60" 'Die Lautstärke des vMix-Inputs mit der
Original-Tonspur, wenn der Übersetzer spricht
dim fadeTimeSpeaking as string = "200" 'Die Zeit mit der die Lautstärke des
Originals abnimmt, wenn der Dolmetscher spricht

dim voicethreshold as string = "0.05" 'Interpreter response threshold, from
0 to 1 (log scale)
' Mik Lautstärke, bei welcher Lautstärke die Regelung anspricht, entspricht
den Werten (nicht der Länge) des Lautstärkereglers: 0.5=50%~-6dB,
0.1=10%~-20dB...
' Die Reaktion funktioniert möglicherweise nicht mit einem einzelnen
Signalimpuls, aber sie funktioniert gut mit einem Signal wie Stimme oder

```

Musik

```
dim checkingIter as integer = 10 'Die Anzahl der Versuche, die Stille des
Interpreter zu überprüfen, bevor wieder die normale Lautstärke des
Originals hergestellt wird (bei 5 wird bei jeder Wortpause das Original
sofort wieder lauter gemacht)
dim checkingIterTime as integer = 200 ' Tdas Intervall (Millisekunden)
zwischen den Versuchen (empfohlen ab 100, still 200)
' Bedeutet die Dauer der Prüfung, ob die Übersetzung still ist und das
Original eingeschaltet werden kann Orig. = checkingIter * checkingIterTime
(Millisekunden)
'---- end of settings----

dim voicemeter as string = ""
dim speaking as boolean = false
dim silence as boolean = false
dim silencechecking as integer = 0
dim muted as boolean = false

do while true

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

voicemeter = (x.SelectSingleNode("//input[@title='"& translateInput
&''']/@meterF1").Value)
if (x.SelectSingleNode("//input[@title='"& translateInput
&''']/@meterF2").Value) > voicemeter
voicemeter = (x.SelectSingleNode("//input[@title='"& translateInput
&''']/@meterF2").Value)
end if
muted = (x.SelectSingleNode("//input[@title='"& translateInput
&''']/@muted").Value)

'API.Function("SetText",Input:="TextInput",SelectedName:="Timer.Text"
,Value:=voicemeter)
'console.writeline(voicemeter)

if voicemeter > voicethreshold And voicemeter.IndexOfAny("[E]".ToCharArray)
= -1 And Not muted

  if silencechecking >= checkingIter '<> 0'!=

    'console.writeline("Speaking")
    silencechecking = 0
'API.Function("SetTextColour",Input:="TextInput",SelectedName:="Timer.Text",
Value:="yellow")
'API.Function("SetText",Input:="TextInput",SelectedName:="Timer.Text"
,Value:="Translation speaks! " + voicemeter)
```

```
        Input.Find(origInput).Function("SetVolumeFade", volumeSpeaking + ","
+ fadeTimeSpeaking)

    end if

    'console.writeline(silencechecking)

else

    if silencechecking < checkingIter
        silencechecking += 1
    end if
    if silencechecking = checkingIter
        'console.writeline("Silence")
'API.Function("SetTextColour",Input:="TextInput",SelectedName:="Timer.Text",
Value:="white")
'API.Function("SetText",Input:="TextInput",SelectedName:="Timer.Text"
,Value:="Translation is silent! " + voicemeter)
        Input.Find(origInput).Function("SetVolumeFade", volumeStandart + ","
+ fadeTimeStandart)
        silencechecking = checkingIter + 1
    end if

    'console.writeline(silencechecking)
end if

sleep(checkingIterTime)

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 Schriftfarbe eines bestimmten Textfeldes (SetTextColour)

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

ATTENTION: if you send the command via HTTP API, replace the # character with %23.

Example: #FF0000 becomes %23FF0000\

You can use HEX rgba values for one color

SAMPLE: rgba(red, green, blue, alpha)

#FF0000 for red 100% visible

#FF00007F for red 50% transparent, 00 = transparent, FF = 100% visible

ACHTUNG: vMix benutzt hier SetTextColour, nicht SetTextColor (mit einem u in Colour)

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

Ändert Schriftfarbe eines bestimmten Textfeldes (SetTextColour)

Wenn ein Textfeld einen bestimmten Wert anzeigt, wird mit diesem Script die Textfarbe geändert. Dies kann die Textfarbe des Textfeldes sein, welche den Wert anzeigt oder irgen ein anderes.

ATTENTION: if you send the command via HTTP API, replace the # character with %23.

Example: #FF0000 becomes %23FF0000\

You can use HEX rgba values for one color

SAMPLE: rgba(red, green, blue, alpha)

#FF0000 for red 100% visible

#FF00007F for red 50% transparent, 00 = transparent, FF = 100% visible

```
dim textvalue as string  
dim numericvalue as integer  
  
Do While True  
    ' get text from title  
    textvalue = Input.Find("Title 0- The Classic  
Blue.gtzip").Text("Headline.Text")  
  
    ' errorhandling, if in the textfield is a non numeric value
```

```
Try
    numericvalue = CInt(textvalue) 'convert the string to a number
Catch ex As Exception
    Console.WriteLine("Error: " & ex.Message)
    ' errorhandling, sets numericvalue to a save value
    numericvalue = -1
End Try

' depending on a value, it changes the color of the text
If numericvalue = 10 Then
    API.Function("SetTextColour", Input:="Title 0- The Classic
Blue.gtzip", Value:="#FF0000", SelectedName:="Description.Text")
ElseIf numericvalue = 20 Then
    API.Function("SetTextColour", Input:="Title 0- The Classic
Blue.gtzip", Value:="#FFFF00", SelectedName:="Description.Text")
ElseIf numericvalue = 30 Then
    API.Function("SetTextColour", Input:="Title 0- The Classic
Blue.gtzip", Value:="#FFFFFF", SelectedName:="Description.Text")
End If

' waita little bit, that the script does not uses all of your cheap
computers power :- )
Sleep(500)
Loop
```

Ändert Shape Farbe eines bestimmten Shapes, z.B. ein Kreis oder ein Rechteck) (SetColor)

ACHTUNG: falls ihr den Befehl via HTTP API schickt, ist das Zeichen # durch %23 zu ersetzen.

Beispiel: #FF0000 wird zu %23FF0000

Es gehen argb Werte für eine Farbe
MUSTER: rgba(red, green, blue, alpha)

#FF000000 für Rot 100% transparent

#FF0000FF für Rot 0% transparent



HEX Werte findest Du hier → https://www.w3schools.com/colors/colors_hexadecimal.asp

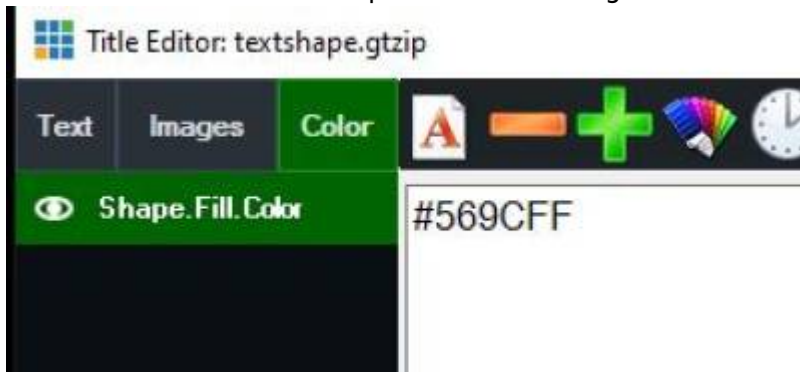
```
API.Function("SetColor",Input:="multiviewer_rotlicht.gtzip",SelectedName:="1.Fill.Color",Value:="#FFFF0000")
```

via HTTP API

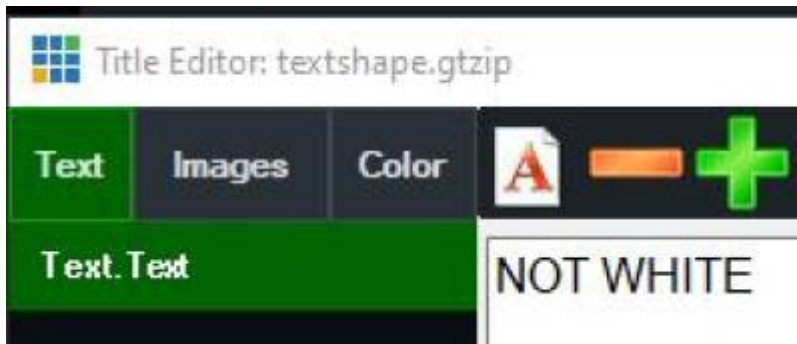
```
http://127.0.0.1:8088/api/?Function=SetColor&Input=clocks.gtzip&SelectedName=background.Fill.Color&Value=%23FFFF00
```

Holt den Farbwert eines bestimmten Shapes, z.B. ein Kreis oder ein Rechteck) (Color)

Liest den Farbwert von Shape.Fill.Color aus / gets the color value of a shape in a gtzip-title:



und schreibt den Wert in das Textfeld Text.Text / and writes the value in the Text.Text field



Titelbeispiel:



[DOWNLOAD GTZIP EXAMPLE](#)

```
do while true 'loop indefinite

'load the XML data from vMix
dim xml as string = API.XML()
dim x as new system.xml.xmldocument
x.loadxml(xml)

' Define the title and text field you're interested in as strings
Dim TITLE As String = "textshape.gtzip"
Dim SHAPE As String = "Shape.Fill.Color"
Dim TEXTFIELD as string ="Text.Text"

' Construct the XPath to locate the color element within the input element
Dim xpath As String = "//input[@title='" & TITLE & "' and color[@name='" &
SHAPE & "']]/color[@name='Shape.Fill.Color']"

' Select the color element using the constructed XPath
Dim colorNode As System.Xml.XmlNode = x.SelectSingleNode(xpath)

' Get the color value from the shape
Dim colorValue As String = colorNode.InnerText

' Output the color value to the textfield in the sample gtzip file
'API.Function("SetText",Input:=TITLE,SelectedName:=TEXTFIELD,Value:=colorVal
ue )
```

```
if colorValue = "#FFFFFF" then 'white
'do something
API.Function("SetText",Input:=TITLE,SelectedName:=TEXTFIELD,Value:="WHITE" )
else
'do something other
API.Function("SetText",Input:=TITLE,SelectedName:=TEXTFIELD,Value:="NOT
WHITE" )
end if
sleep(1000) 'waits one second to loop
loop
```

Zahlenwert auf Tastendruck erhöhen, verringern

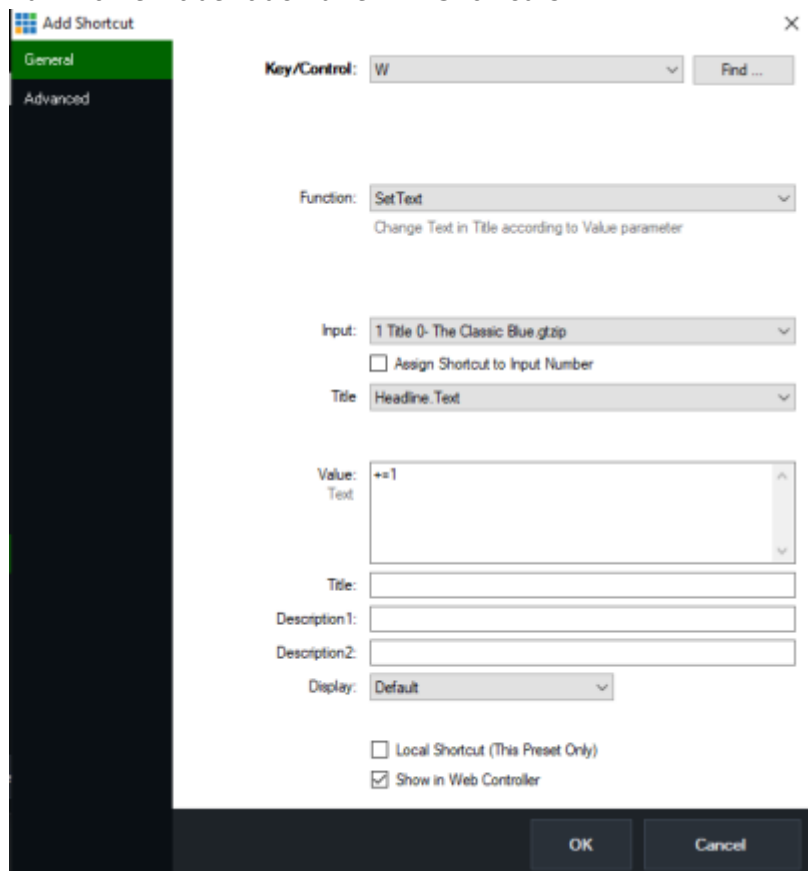
Addieren +=1

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.T
ext",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.T
ext",Value:="transparent")
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg21.T
ext",Value:="transparent")
API.Function("SetTextColour",Input:="whoisonair.gtzip",SelectedName:="hg31.T
ext",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

```

Holt Filename (inkl. Pfad eines Logos aus einem Titel

In diesem Beispiel hat es im Titel in dem Eingang 5, 2 Logos.

Das Ergebnis sind 2 strings wie folgt:

C:\vMix\4ball\Futsal\Logos\SUI.png

C:\VMIX\4ball\Futsal\Logos\SVN.png

```

dim logo1 as string = ""
dim logo2 as string = ""

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

logo1 = (x.SelectSingleNode("//input[@number=5]/image[1]").InnerText)
logo2 = (x.SelectSingleNode("//input[@number=5]/image[2]").InnerText)

```

```
console.writeline(logo1)
console.writeline(logo2)
```

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
```

Holt die Namen und Nummern aller vMix Caller und speichert diese in eine Datei (4K Version mit 4 callern)

```
'Fill in complete path to folder, end with \
dim FILEPATH as string = "C:\User\Example\Folder\"
dim FILENAME as string = "Caller_Log.txt"

' Leave the rest untouched

dim FILE as string = FILEPATH & FILENAME

dim callers() as string = {"caller1","caller2","caller3","caller4"}
dim nodes() as string = {"//input[@number=1]", "//input[@number=2]",
"//input[@number=3]", "//input[@number=4]"}

dim i as integer = 0

do while true
    dim xml as string = API.XML()
```

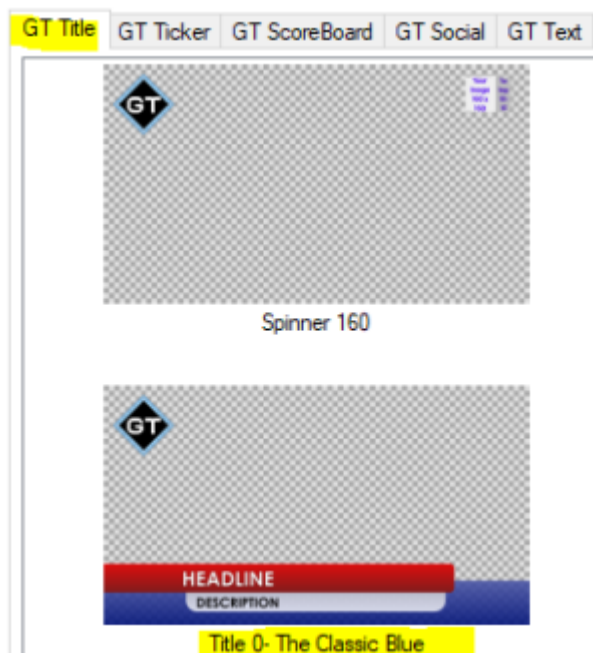
```

dim x as new system.xml.xmldocument
x.loadxml(xml)
do while i < 4
    dim node as string = nodes(i)
    dim caller as string = callers(i)
    dim name as string = (x.SelectSingleNode(node)).InnerText
    If name <> caller:
        Dim callerid As string = (i+1).ToString()
        Dim timeStamp As DateTime = DateTime.Now
        Dim callertxt as string = "- New caller at ID" & " " & callerid
        & " - "& name
        Dim writenew as string = timeStamp & " " & callertxt
        System.IO.File.AppendAllText(FILE, writenew +
Environment.NewLine)
        Console.writeline(writenew)
        callers(i) = name
    End If
    i = i+1
loop
i = 0
sleep(1000)
loop

```

läuft die Aufzeichnung?

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



```

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
```

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)

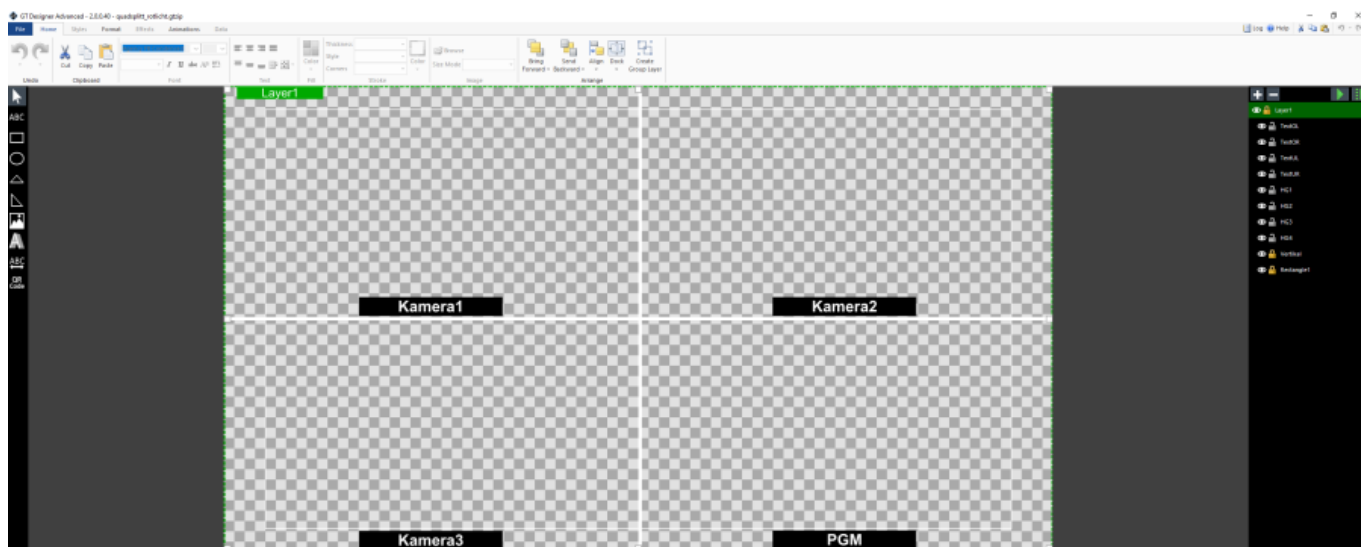
'Aufnahmezeit in die Datei schreiben
System.IO.File.AppendAllText(filename,
string.format("{0:00}:{1:00}:{2:00}"&Environment.NewLine, Hour, Minute,
Second))
```

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",SelectedName
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",SelectedName
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",SelectedName
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 Scriptes aus einem Script braucht Zeit. Also immer nach einer komplexeren Aktion ein `sleep(200)` setzen.

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

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

Ich bin nicht sicher, ob diese Variante in vMix unter 27 läuft, falls nein hat es unten noch ein älteres Beispiel mit `WebRequest`, welches läuft. Das Beispiel verwendet auch `WebRequest`, welches ein Grundbestandteil von VB Net ist, leider hat vMix das modernere `HttpClient` nicht implementiert. Ich verwende im Code aber `Using/End Using`. Dieses Verfahren löscht die verwendeten Elemente am Ende vollständig durch `End Using`.

Der Code schaltet `Input2` auf `Overlay1` am Computer mit der IP `192.168.10.102` ein/aus.(toggle)

```
' vMix kommando (url)
Dim url As String =
"http://192.168.10.102:8088/api/?Function=OverlayInput1&Input=2"

' erstellt WebRequest
Dim request As System.Net.WebRequest = System.Net.WebRequest.Create(url)

' sendet anfrage und erhält antwort
Using response As System.Net.WebResponse = request.GetResponse()
    ' stream aus der antwort
    Using stream As System.IO.Stream = response.GetResponseStream()
        'streamReader verwenden, um antwortstream zu lesen und antwort in
        console schreiben
        Using reader As New System.IO.StreamReader(stream)
            Dim responseContent As String = reader.ReadToEnd()
            Console.WriteLine(responseContent)
        End Using
    End Using
End Using
```

Das gleiche mit Fehlerbehandlung

```
' vMix kommando (url)
Dim url As String =
"http://192.168.10.102:8088/api/?Function=OverlayInput1&Input=2"

Try
    ' erstellt WebRequest
    Dim request As System.Net.WebRequest = System.Net.WebRequest.Create(url)
```

```

' sendet anfrage und erhält antwort
Using response As System.Net.WebResponse = request.GetResponse()
' stream aus der antwort
Using stream As System.IO.Stream = response.GetResponseStream()
' streamReader verwenden, um antwortstream zu lesen und antwort in
console schreiben
Using reader As New System.IO.StreamReader(stream)
Dim responseContent As String = reader.ReadToEnd()
Console.WriteLine(responseContent)
End Using
End Using
End Using

Catch ex As System.Net.WebException
' netzwerk fehler
Console.WriteLine("Netzwerkfehler: " & ex.Message)
If ex.Response IsNot Nothing Then
' zusatzinfos von webresponse
Using exResponse As System.Net.HttpWebResponse =
DirectCast(ex.Response, System.Net.HttpWebResponse)
Console.WriteLine("HTTP Status Code: " &
exResponse.StatusCode)
End Using
End If
Catch ex As System.IO.IOException
' fehler stream lesen
Console.WriteLine("IO-Fehler: " & ex.Message)
Catch ex As Exception
' unbekante fehler
Console.WriteLine("UnbkannterFehler: " & ex.Message)
End Try

```

Dieses ältere Beispiel schaltet Input2 auf Overlay1 am Computer mit der IP 192.168.10.102 ein/aus(toggle).

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

```

Dim client =
WebRequest.Create("http://192.168.10.102:8088/api/?Function=OverlayInput1&In
put=2")
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)
```

Sichert alle Scripte als Textfile in einen vorbestimmten Ordner

Dieses vMix-Script sichert alle aktiven Scripts in vMix als einzelne Textfiles in einen vorbestimmten Ordner.

Damit kann von den benutzten scripts ein Backup gemacht werden. Diese Funktion ist in vMix nicht vorgesehen. Man kann nur alle Scripts und Shortcuts als Ganzes exportieren. Beim Import werden dann eventuell später dazugekommene Scripts gelöscht.

WICHTIG, nach einem Update muss der Pfadauf die neue Version angepasst werden!

```
'Suche dein User Config im Appdata\local\studiocoast ordner
' bei mir sieht das so aus:
'
C:\Users\Peter_PC\AppData\Local\StudioCoast_Pty_Ltd\VMix64.exe_Url_pnmwemdfj
\24.0.0.63\user.config
' WICHTIG, nach einem Update muss der Pfad auf die neue Version angepasst
werden!
'ab hier beginnt das script
Dim vMixConfig as string =
"C:\Users\Peter_PC\AppData\Local\StudioCoast_Pty_Ltd\VMix64.exe_Url_pnmwemdf
j\24.0.0.63\user.config"

'Bestimme, wo die einzelnen Scripts gespeichert werden sollen
Dim path as string = "D:\Myscripts\"

'falls der Pfad nicht existiert, wird er angelegt
If(Not System.IO.Directory.Exists(path)) Then
    System.IO.Directory.CreateDirectory(path)
End If

'xml initialisieren
Dim document As XmlDocument = New XmlDocument()
'lädt das config xml file
document.Load(vMixconfig)

'holt sich den inneren xml-String für die Skripte
dim innerxml as string =
(document.SelectSingleNode("//setting[@name='Scripts']").InnerText)
'lädt die enthaltenen skripte
Dim doc As New XmlDocument()
doc.LoadXML(innerxml)

Dim nodeList As XmlNodeList = (doc.SelectNodes("//Script"))

'speichert die skripte im vorgesehenen Ordner ab
For Each node As XmlNode In nodeList
    Dim thefile As System.IO.Streamwriter
    'Console.WriteLine(node("Name").InnerText )
    thefile = My.Computer.FileSystem.OpenTextFileWriter(path +
node("Name").InnerText + ".txt", False)
    thefile.WriteLine(node("Code").InnerText)
```

```

    thefile.Close()
next

```

crlf (carriage return) aus Textstring entfernen

```
string1= string1.Replace(Environment.NewLine,String.Empty)
```

ist ein Input gestoppt? (state value)

'Is a input Running or on Pause retrieved from the API xml

```

dim xml = API.XML()
dim x as new system.xml.xmldocument
x.loadxml(xml)
dim StateOf = (x.SelectSingleNode("//input[@type='Replay']/@state").Value)
'oder
'dim StateOf = (x.SelectSingleNode("//input[@number='10']/@state").Value)

if StateOf = true
' mach etwas
else
' mach etwas anderes
end if

```

Dynamicvalue

Ab Version 25 hat vMix eine „Art“ globalen Variablenspeicher. Dieser nennt sich Dynamicvalue. Es gibt 4 davon. Mit der Funktion SetDynamicValue1 wird dieser Wert in die API Datenbank geschrieben (steht nicht direkt Global zur Verfügung)

```
API.Function("SetDynamicValue1", Value:="Dieser Text wird in seiner ganzen Schönheit als Dynamic1 abgelegt")
```

Auslesen wie folgt:

```

'-----holt Dynamicvalue1 als verwertbaren String (VARIABLE) zum
weiterverarbeiten -----
'-----Dynamicvalue1 kann als globale Variable verwendet werden -----
----
'-----es geht nur dieser Weg über die API, da die Variable nur im XML File
abgelegt wird---
dim VARIABLE as string
dim Dynamic1Node As XmlNode
dim VmixXML as new system.xml.xmldocument

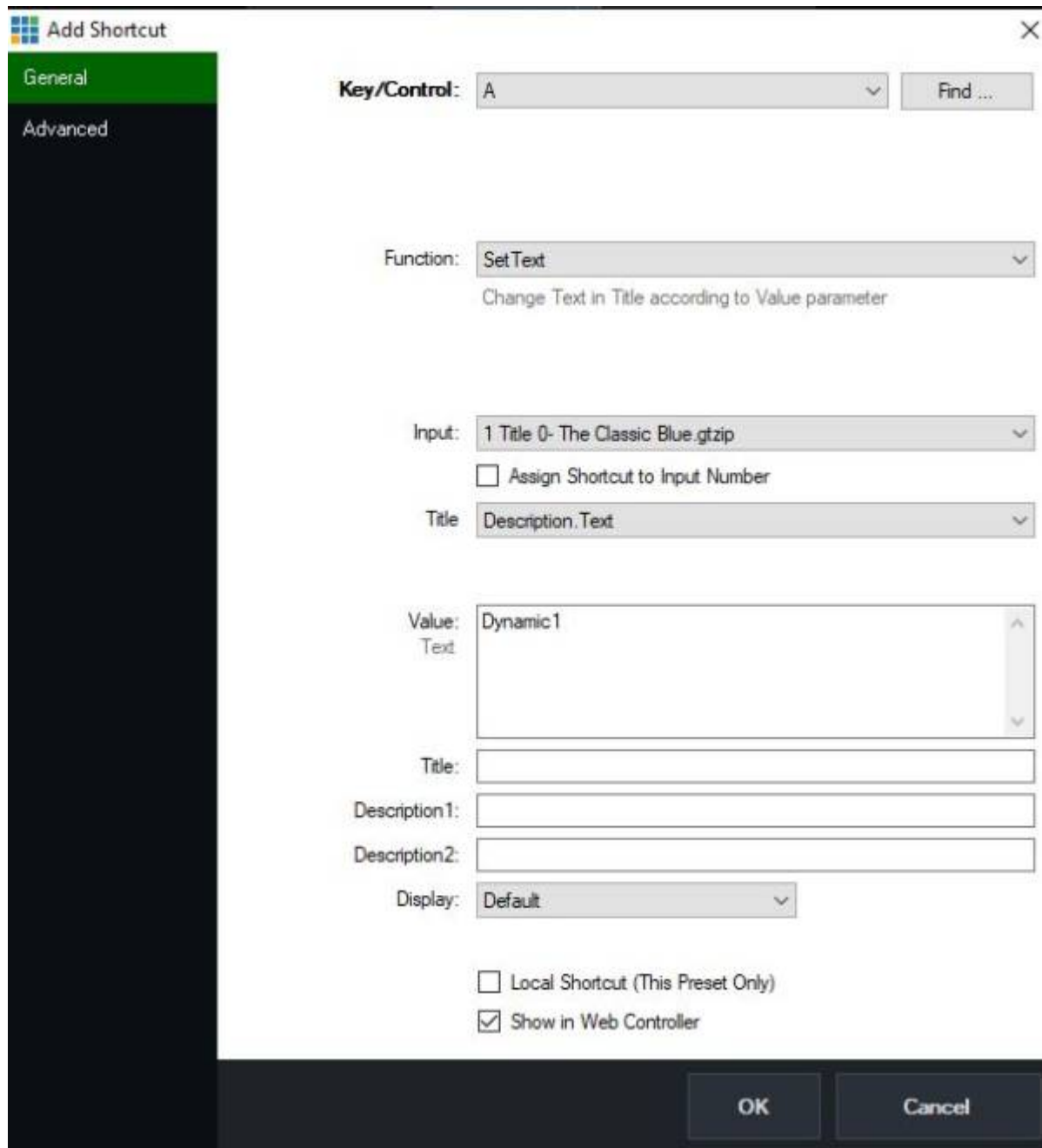
```

```
VmixXML.loadxml(API.XML)  
Dynamic1Node= VmixXML.selectSingleNode("/vmix/dynamic/value1")  
VARIABLE = Dynamic1Node.InnerText  
console.writeline(VARIABLE)
```

Somit liesse sich ein Titel mit dieser Variable füllen, mit einem Script anhang des Wertes eine Aktion auslösen etc.

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

Oder direkt mit einem Shortcut:



DynamicInput

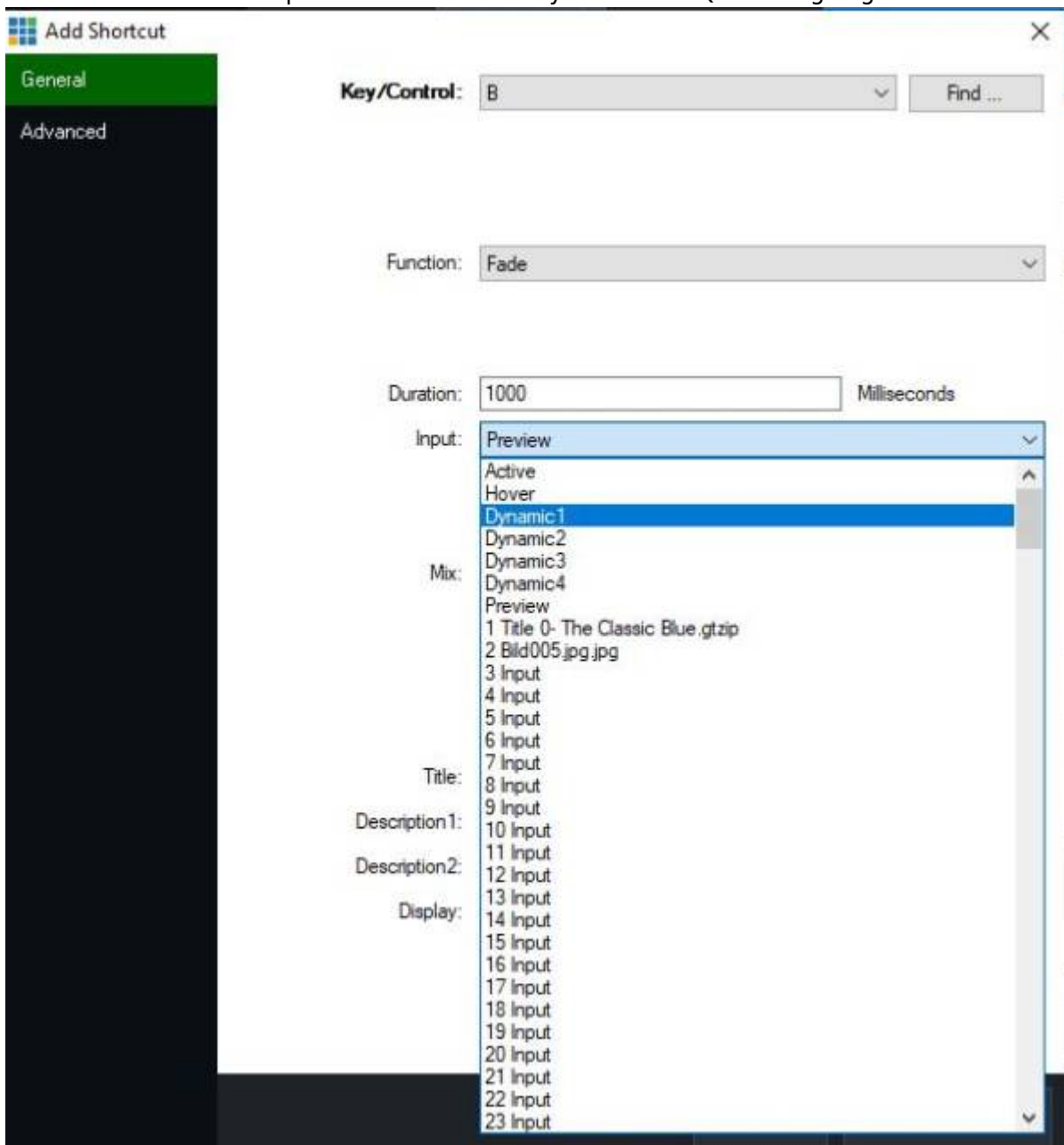
Gleichermassen wie Dynamicvalue funktioniert auch DynamicInput. Es gibt auch 4 davon. Mit der Funktion SetDynamicInput1 wird dieser Wert in die API Datenbank geschrieben (steht nicht direkt Global zur Verfügung)

```
API.Function("SetDynamicInput1", Value:="My input")
```

oder

```
API.Function("SetDynamicInput1", Value:=1)
```

Danach kann in der Dropdownliste der Wert Dynamic1 als Quelle angelegt werden.



Es gibt ein kurzes Video von vMix dazu, welches dieses gut erklärt. [vMix Video Dynamic Shortcuts](#)

String Funktionen

Holt alle Zeichen vor einem bestimmten Character in einem string

Holt allen Text aus einem String bis zum Zeichen <.

```
Dim leftPart As String = string1.Split("<")(0)
```

oder

```
Dim leftPart As String
Dim index As Integer = string1.IndexOf("<")
If index <> -1 Then
    leftPart = string1.Substring(0, index)
Else
    leftPart = string1
End If
```

oder

```
Dim leftPart As String
Dim index As Integer = string1.IndexOf("<")
If index <> -1 Then
    leftPart = string1.Remove(index)
Else
    leftPart = string1
End If
```

Ermittelt die ersten 3 Zeichen eines Strings

```
Dim originalstring As String
originalstring = "Peter Paul und Marry haben 455 Franken"

Dim first3 As String
first3 = originalstring.Substring(0, 3)
' die Variable first3 enthält string "Pet"
```

Ermittelt die letzten 6 Zeichen eines Strings

```
Dim originalstring As String
originalstring = "Peter Paul und Marry haben 455 Franken"
```

```
Dim last6 As String
last6 = originalstring.Substring(originalstring.Length - 6)
' die Variable last6 enthält "Franke"
```

Ermittelt die Länge eines Strings

```
Dim originalstring As String
originalstring = "Peter Paul und Marry haben 455 Franken"

Dim length As Integer
length = originalstring.Length
' die Variable length enthält 32 (Integer)
```

Ersetze Marry mit Lisa

```
Dim originalstring As String
originalstring = "Peter Paul und Marry haben 455 Franken"

Dim replaced1 As String
replaced1 = originalstring.Replace("Marry", "Lisa")
' die Variable replaced1 enthält "Peter Paul und Lisa haben 455 Franken"
```

Ersetze Peter und Marry mit John und Lisa

```
Dim originalstring As String
originalstring = "Peter Paul und Marry haben 455 Franken"

Dim replaced2 As String
replaced2 = originalstring.Replace("Peter", "John").Replace("Marry", "Lisa")
' die Variable replaced2 enthält "John Paul und Lisa haben 455 Franken"
```

Alle Zeichen links vom Komma

```
Dim originalstring As String
originalstring = "Peter, Paul und Marry haben: 455 Franken"

Dim leftOfComma As String
leftOfComma = originalstring.Substring(0, originalstring.IndexOf(","))
' die Variable leftOfComma enthält "Peter"
```

Alle Zeichen rechts vom Doppelpunkt

```
Dim originalstring As String
originalstring = "Peter, Paul und Marry haben: 455 Franken"

Dim rightOfColon As String
rightOfColon = originalstring.Substring(originalstring.IndexOf(":") + 1)
' die Variable rightOfColon enthält " 455 Franken"
```

Alle Zeichen links und rechts vom Komma

```
Dim originalstring As String
originalstring = "Peter, Paul und Marry haben: 455 Franken"

Dim leftOfComma As String
Dim rightOfComma As String

leftOfComma = originalstring.Substring(0, originalstring.IndexOf(","))
rightOfComma = originalstring.Substring(originalstring.IndexOf(",") + 1).Trim()

' die Variable leftOfComma enthält "Peter"
' die Variable rightOfColon enthält "Paul und Marry haben: 455 Franken"
```

separiere alle Namen in einzelne variablen

Dieser Codeabschnitt nimmt den ursprünglichen String `originalstring` und teilt ihn in ein Array von Substrings auf, indem er das Komma als Trennzeichen verwendet. Da der zweite Teil des ursprünglichen Strings die Namen enthält, greift der Code auf das zweite Element des Arrays zu, indem er (1) verwendet. Danach wird der zweite Teilstring, der die Namen enthält, mit dem Wort „und“ als Trennzeichen in ein weiteres Array von Substrings aufgeteilt. Das Ergebnis ist ein String-Array `names`, das die einzelnen Namen enthält.

In diesem Beispiel wird der ursprüngliche String „Peter, Paul und Marry haben: 455 Franken“ in drei Namen („Peter“, „Paul“ und „Marry“) aufgeteilt und in das `names`-Array gespeichert.

```
Dim originalstring As String
originalstring = "Peter, Paul und Marry haben: 455 Franken"

Dim names As String() = originalstring.Split(",")(1).Split("und")
Dim name1 As String = names(0).Trim()
Dim name2 As String = names(1).Trim()
Dim name3 As String = names(2).Trim()

' die Variable name1 enthält "Peter"
' die Variable name2 enthält "Paul"
```

```
' die Variable name3 enthält "Marry"
```

Teilt alle einzelnen Wörter, inklusive Satzzeichen, in ein Array und sucht dann in welchem Array Paul steht

In diesem Beispiel wird der ursprüngliche String „Peter, Paul und Marry haben: 455 Franken“ in ein Array von Wörtern aufgeteilt, indem er Leerzeichen, Kommas und Doppelpunkte als Trennzeichen verwendet. Das Ergebnis ist ein String-Array words, das alle Wörter im ursprünglichen String enthält. Dann wird nach dem Wort „Paul“ im words-Array gesucht, indem die Methode Array.IndexOf() verwendet wird. Wenn das Wort gefunden wird, wird die Position des Wortes im Array angezeigt. Andernfalls wird angezeigt, dass das Wort nicht gefunden wurde.

```
Dim originalstring As String
originalstring = "Peter, Paul und Marry haben: 455 Franken"

' Zerlegen der Zeichenkette in ein Array von Wörtern
' trennzeichen kann man ergänzen/ändern "- "c, "."c, etc.
Dim words As String() = originalstring.Split(New Char() {" "c, ","c, ":"c},
StringSplitOptions.RemoveEmptyEntries)

' Suche das Wort "Paul"
Dim paulIndex As Integer = Array.IndexOf(words, "Paul")
If paulIndex <> -1 Then
    Dim foundWord As String = words(paulIndex)
    MsgBox("Paul gefunden im Index " & paulIndex & ", das gefundene Wort ist " & foundWord)
Else
    MsgBox("Paul nicht gefunden")
End If
```

Ersetze einen bestimmten Textteil durch einen anderen

```
Dim originalstring As String
originalstring = "0xff0000"

' Ersetze "0x" mit ""
Dim newString As String = originalstring.Replace("0x", "")
' die Variable newString enthält "ff0000"
```

Lösche alle Leerzeichen am Anfang und am Ende einer Zeichenkette

```
Dim originalstring As String
originalstring = " Peter, Paul und Marry haben: 455 Franken "
```

```
Dim newString As String = originalstring.Trim()  
' die Variable newString enthält "Peter, Paul und Marry haben: 455 Franken"
```

ersetzt Anführungszeichen in einem String

der zu bearbeitende String ist in einem Textfeld(Headline.Text) eines Titels (Title.gtzip), Gross-/Kleinschreibung immer beachten. Es wird ein Anführungszeichen („) durch ein einfaches Anführungszeichen (') ersetzt.

```
Dim string1 as String = (Input.Find("Title.gtzip").Text("Headline.Text"))  
Dim string2 As String = string1.Replace( """" , "'")
```

Holt den linken Teil eines Stings, vor einem bestimmten Zeichen

Originalstring:

Currently in Lexington, VA: 25 °F and Mostly Cloudy <img
src=\"https://vortex.accuweather.com/phoenix2/images/common/icons/38_31x31.gif„>

Zeigt alles vor dem Zeichen "<„

```
Dim leftPart As String = string1.Split("<")(0)
```

Currently in Lexington, VA: 25 °F and Mostly Cloudy

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 Filenameas 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)

```

kopiert 10 Videos aus einem Verzeichnis in ein anderes

Hat man Werbung in verschiedenen Sprachen, kann man die Videos einer Sprache ins Playlist Verzeichnis kopieren, ohne an der Playlist etwas zu ändern.

```

' Französisch

Dim Sprache as String = "f"

'Verzeichnis in dem die französischen Werbevideos abgelegt sind
'ein französisches Video in diesem Verzeichnis muss dann heissen:
E:\media\Videos_f\Video1.mp4

Dim V1 As String = "E:\media\Videos_" + Sprache + "\Video1.mp4"
Dim V2 As String = "E:\media\Videos_" + Sprache + "\Video2.mp4"
Dim V3 As String = "E:\media\Videos_" + Sprache + "\Video3.mp4"
Dim V4 As String = "E:\media\Videos_" + Sprache + "\Video4.mp4"
Dim V5 As String = "E:\media\Videos_" + Sprache + "\Video5.mp4"
Dim V6 As String = "E:\media\Videos_" + Sprache + "\Video6.mp4"
Dim V7 As String = "E:\media\Videos_" + Sprache + "\Video7.mp4"
Dim V8 As String = "E:\media\Videos_" + Sprache + "\Video8.mp4"
Dim V9 As String = "E:\media\Videos_" + Sprache + "\Video9.mp4"
Dim V10 As String = "E:\media\Videos_" + Sprache + "\Video10.mp4"

```

```
'Verzeichnis, aus dem vMix die Playlist erstellt
Dim V1P As String = "E:\media\Videos\Video1.mp4"
Dim V2P As String = "E:\media\Videos\Video2.mp4"
Dim V3P As String = "E:\media\Videos\Video3.mp4"
Dim V4P As String = "E:\media\Videos\Video4.mp4"
Dim V5P As String = "E:\media\Videos\Video5.mp4"
Dim V6P As String = "E:\media\Videos\Video6.mp4"
Dim V7P As String = "E:\media\Videos\Video7.mp4"
Dim V8P As String = "E:\media\Videos\Video8.mp4"
Dim V9P As String = "E:\media\Videos\Video9.mp4"
Dim V10P As String = "E:\media\Videos\Video10.mp4"
```

Try

```
File.Copy (V1, V1P, True)
File.Copy (V2, V2P, True)
File.Copy (V3, V3P, True)
File.Copy (V4, V4P, True)
File.Copy (V5, V5P, True)
File.Copy (V6, V6P, True)
File.Copy (V7, V7P, True)
File.Copy (V8, V8P, True)
File.Copy (V9, V9P, True)
File.Copy (V10, V10P, True)
Catch iox As IOException
    Console.WriteLine(iox.Message)
End Try
```

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 Beistpiel 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, welch 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 Beistpiel 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
```

vMix Befehle in einer bestimmten Zeitabfolge vorgegebener Zeiten

This script keeps a list of times; when such a time is reached, a vMix command is executed. In this example, Input1 is displayed on Overlay 1.

```
' in the following code line, you put your needed times in the 24h format,
with quotes, separated by a comma
' you can put as many times in, as you need
' this could be different, when your windows is set to 12h format, tha am/pm
thing???
Dim zeiten As String() = {"09:53", "09:54", "09:55", "09:56", "09:57",
"09:58"}

'this is to check, if a new day has come. if this is true, the already
executed times will be cleared
Dim lastExecutionDate As Date = Date.Today

' this array holds the already executed times
Dim executed As Boolean() = New Boolean(zeiten.Length - 1) {}

'loop starts here
While True
    Dim currentTime As DateTime = DateTime.Now

    'check, if a new day has come, if this is true, the already executed
times will be cleared
    If lastExecutionDate <> Date.Today Then
        executed = New Boolean(zeiten.Length - 1) {}
        lastExecutionDate = Date.Today
    End If

    'check if the actual time is in your array, and, if not already
executed, execute the vMix comand or commands
```

```

Dim currentFormattedTime As String = currentTime.ToString("HH:mm")
For i As Integer = 0 To zeiten.Length - 1
    If zeiten(i) = currentFormattedTime AndAlso Not executed(i) Then
        ' here the vMix command will be executed, when a new time is
        true, you could add more commands here
        ' this example puts overlay 1 online, with Input1, please change
        this to your needs
        API.Function("OverlayInput1In", Input:="1")
        executed(i) = True ' mark the time as executed
        Exit For
    End If
Next
Sleep(200)
End While

```

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“)
```

SetImageVisible kann ein Bild ein-oder ausschalten (**toggle**)

SetImageVisibleON kann ein Bild einschalten

SetImageVisibleOFF kann ein Bild ausschalten

Auch im GT-Titler erzeugte Shapes wie Rectangles können mittels des Alpha Kanals der Farbe ein- und ausgeschaltet werden.

MUSTER: rgba(red, green, blue, **alpha**)

[Beispieltitel hier:](#)

visibledemo.gtzip

```
dim visible as boolean = false
do while true

if visible = false
API.Function("SetTextVisibleON",Input:="visibledemo.gtzip",SelectedName:="TextBlock1.Text")
API.Function("SetImageVisibleON",Input:="visibledemo.gtzip",SelectedName:="Image1.Source")
  API.Function("SetColor",Input:="visibledemo.gtzip",SelectedName:="Rectangle1.Fill.Color",Value:="#FF0000FF")
  visible = true
else
API.Function("SetTextVisibleOFF",Input:="visibledemo.gtzip",SelectedName:="TextBlock1.Text")
API.Function("SetImageVisibleOFF",Input:="visibledemo.gtzip",SelectedName:="Image1.Source")
  API.Function("SetColor",Input:="visibledemo.gtzip",SelectedName:="Rectangle1.Fill.Color",Value:="#FF000000")
  visible = false
end if

sleep(500)

loop
```

Text ein/aus aufgrund eines bestimmten Wertes, Resultats etc.

Macht Text oder Bilder unsichtbar, wenn in einem Titel ein Textfeld einen bestimmten Wert hat. Dieses Beispiel verwendet den gleichen Titel wie im obigen Beispiel, visibledemo.gtzip und schaltet Text und Bild im visibledemo.gtzip ein und aus, aufgrund eines Wertes vom Titel „Title 0- The Classic Blue.gtzip“, im Feld „Headline.Text“.

```
dim a as string

do while true

'fills variable a with the content of Headline.Text
a= Input.Find("Title 0- The Classic Blue.gtzip").Text("Headline.Text")

'if the content is exactly "a" then the text is visible, any other value
hides the text
if a = "a" then
API.Function("SetTextVisibleON",Input:="visibledemo.gtzip",SelectedName:="TextBlock1.Text")
API.Function("SetImageVisibleON",Input:="visibledemo.gtzip",SelectedName:="Image1.Source")
  API.Function("SetColor",Input:="visibledemo.gtzip",SelectedName:="Rectangle1.Fill.Color",Value:="#FF0000FF")
else
API.Function("SetTextVisibleOFF",Input:="visibledemo.gtzip",SelectedName:="TextBlock1.Text")
API.Function("SetImageVisibleOFF",Input:="visibledemo.gtzip",SelectedName:="Image1.Source")
  API.Function("SetColor",Input:="visibledemo.gtzip",SelectedName:="Rectangle1.Fill.Color",Value:="#FF000000")
end if

sleep(500)

loop
```

MIX Funktionen

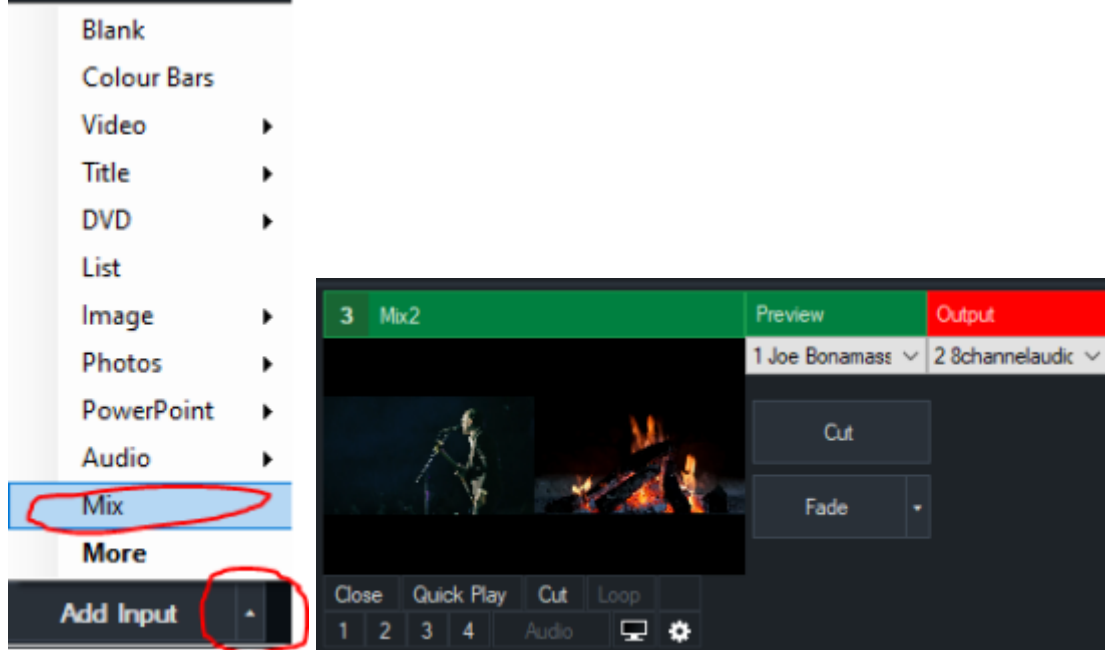
Mixer Input

vMix hat noch (ab Version 26) 16 kleine Mixer Inputs, welche auch gut für verschiedene scriptgesteuerte Funktionen verwendet werden können. Der Vorteil der zusätzlichen Mischer ist, dass jeder Eingang auch geblendet werden kann. (Die Mischer sind nur ab der 4K Version verfügbar)

Hier ist wieder Mix=0 der OUTPUT/PGM, Mix=1 der erste Input-MIX,

welcher dann aber im kleinen Vorschauenfenster mit Mix2
angeschrieben ist.... Verwirrend, ist aber so.

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 (0 im
Scripting = 1 in Shortcut)
```

oder:
(Mix0 ist Programm)

```
http://192.168.10.102:8088/api/?Function=Fade&Input=2&Mix=1
```

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.

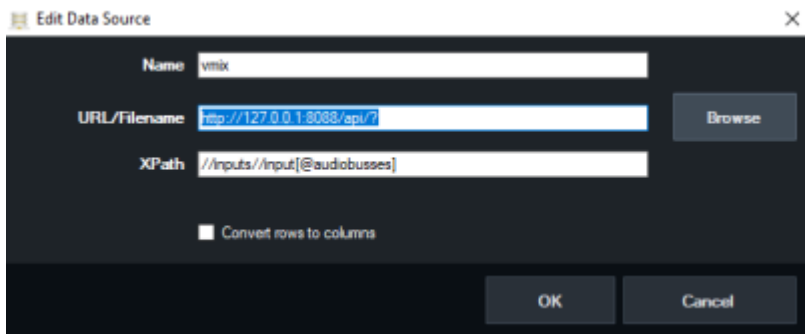
http://xmltoolbox.appspot.com/xpath_generator.html (liefert ein Ergebnis basierend auf einer Auswahl)

<https://www.easycodewall.com/XPathUtility.jsp> (gibt eine ganze Liste möglicher xpath's 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

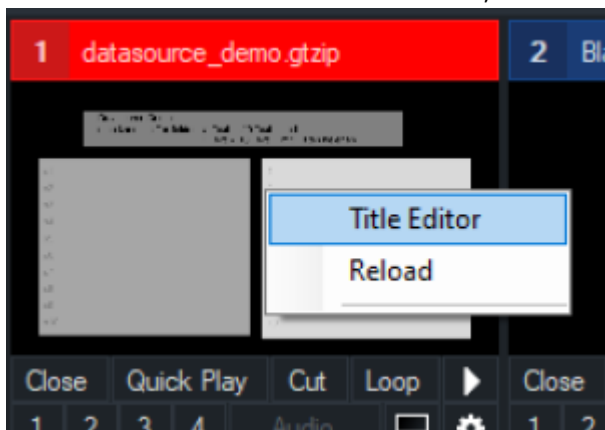


API von vMix als Datasource nutzen

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.gtzip

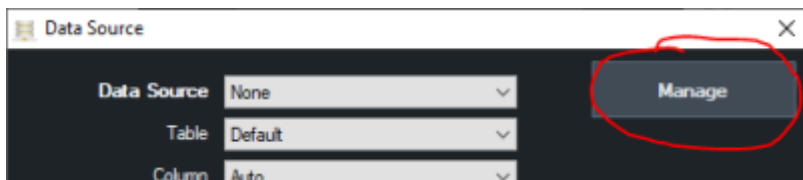
1. Rechte Maustaste ins Titelfenster, Title Editor



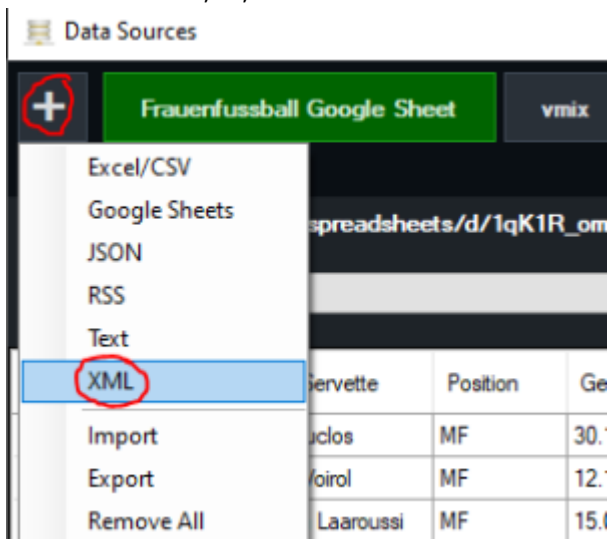
2. Data Source



3. Manage



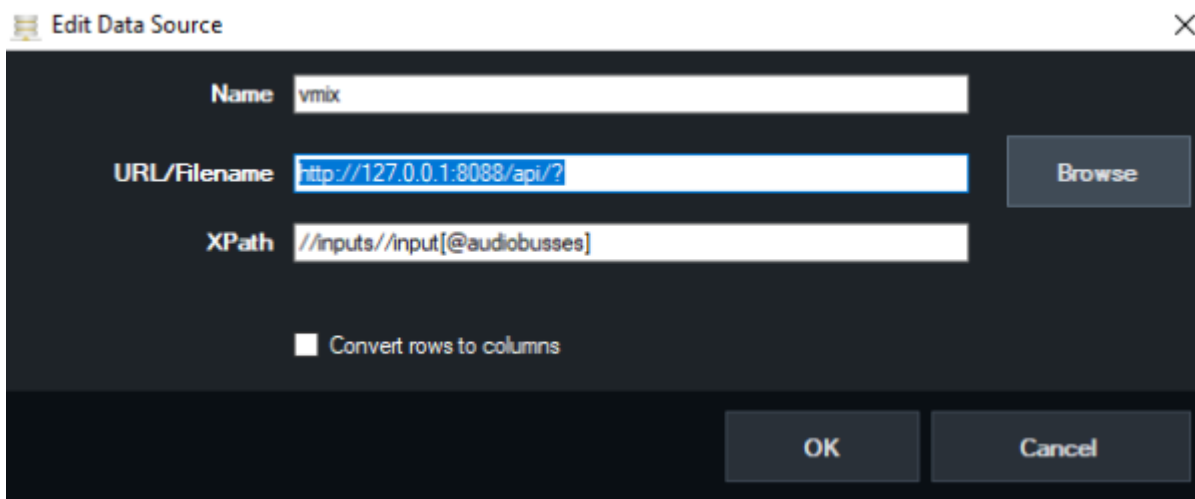
4. Data Source, +, XML



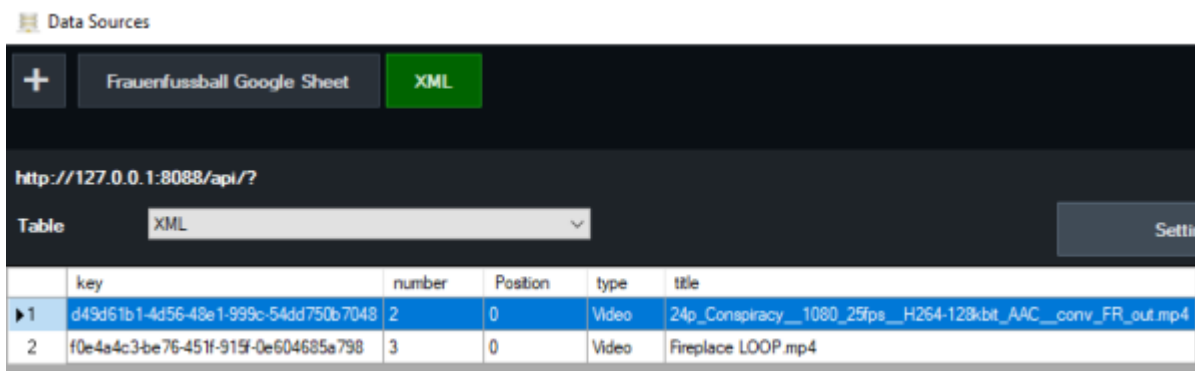
5. Name,

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

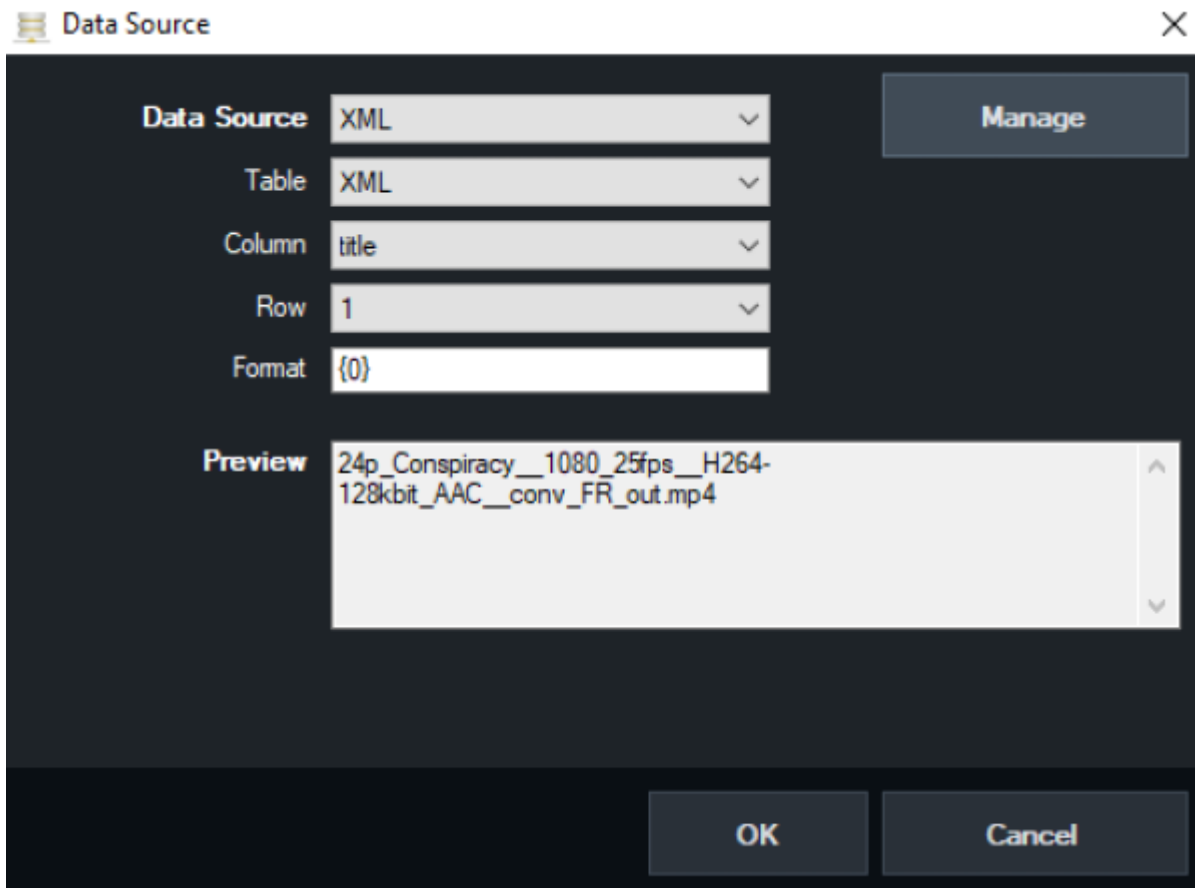
Dann OK drücken.



6. Das sollte dann so aussehen:



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



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

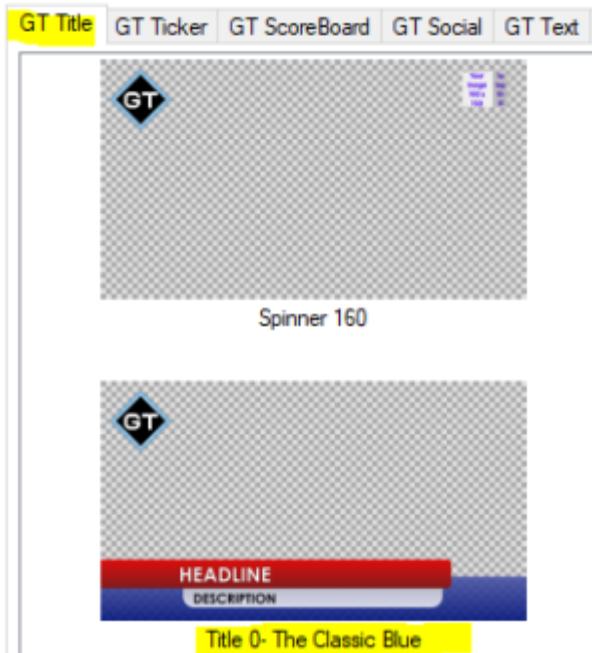
```
API.Function("StreamingSetUsername","Input:=1", "USER99")
```

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.gzip**



```

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")
```

external Output beim Starten korrekt initialisieren (AUTOSTART)

Da Blackmagic Decklink Karten in vMix nicht korrekt initialisieren, MUSS man external Output zur Sicherheit 3 x Ein- und Auschalten. vMix ist dieses Problem bekannt, ist aber offensichtlich nicht in der Lage das Problem auf Ihrer Seite zu lösen.

Dazu braucht es in JEDEM Projekt ein Script und einen Webbrowser Eingang. Startet man das Projekt, wird die Webseite einmal aufgerufen. Da via Webbrowser auch ein vMix API Command geschickt werden kann, löst dieser Befehl nun das Script aus.

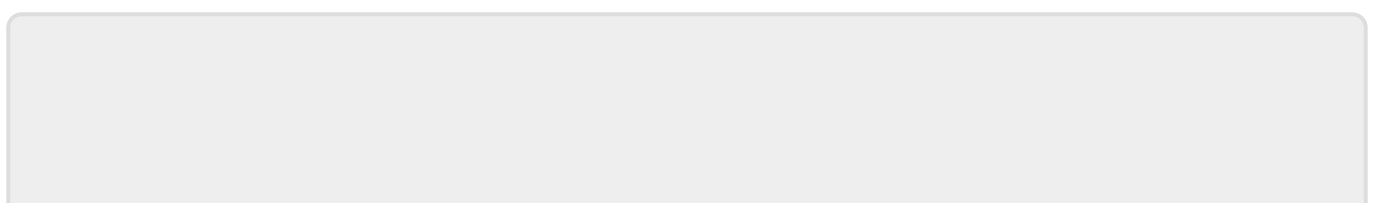
Der Webbrowser Eingang hat folgende URL:

```
http://127.0.0.1:8088/api/?Function=ScriptStart&Value=startexternal
```

Auf dem Computer muss folgendes Script mit dem Namen „startexternal“ vorhanden sein:

```
API.Function("StartExternal")
sleep (1000)
API.Function("StopExternal")
sleep (1000)
API.Function("StartExternal")
sleep (1000)
API.Function("StopExternal")
sleep (1000)
API.Function("StartExternal")
```

Mit diesem Trick kann nahezu jede beliebige Funktion als Autostart Befehl ausgeführt werden.



From:

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

Permanent link:

<https://www.tvcrew.ch/vmix/doku.php?id=scripting>

Last update: **2025/12/05 10:51**

