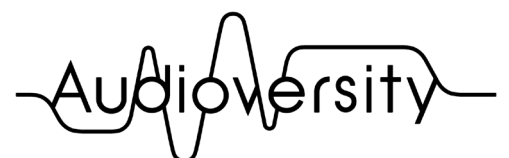




# QLab セットアップガイド for CL/QL/TF シリーズ



**CL** SERIES  
**QL** SERIES  
**TF** SERIES



by Yamaha Pro Audio

## 目次

1.	はじめに.....	3
2.	QLab から制御可能な機能およびパラメーター.....	3
3.	使用機材.....	4
4.	スクリプトテンプレートのダウンロード.....	5
5.	コンソールと Mac の接続.....	6
6.	IP アドレスの設定.....	7
7.	スクリプトテンプレートの設定.....	8
8.	QLab のプログラミング(CL/QL シリーズの場合).....	10
8-1.	シーンのリコール.....	12
8-2.	各入力チャンネルのオンオフ、パン/フェーダーレベルの設定.....	13
9.	QLab のプログラミング (TF シリーズの場合).....	16
9-1.	シーンのリコール.....	17
9-2.	各入力チャンネルのオンオフ、パン/フェーダーレベルの設定.....	19

## 履歴

2019 年 8 月      Version 1.0 発行

当ドキュメントに掲載されているヤマハの商品の名称等はヤマハの商標または登録商標です。

本サイトで使用されている QLab の写真、画像、スクリーンショット等は Figure53 社の商標または登録商標です。

## 1. はじめに

このガイドでは、業界標準のショーコントロールソフトウェアである Figure53 社の QLab から CL/QL/TF シリーズをリモートコントロールするための接続、設定方法について述べられています。また、弊社ウェブサイトからダウンロード可能なサンプルファイルの具体的な使用方法についても記載してあります。CL/QL/TF シリーズのリモートコントロールはコマンド形式のテキストをミキサーに送ることで制御されますので、Windows PC や Mac のターミナルを用いることで操作は可能ですが、このドキュメントでは特に QLab から Python スクリプトを介してコマンドを送信する方法について説明します。

## 2. QLab から制御可能な機能およびパラメーター

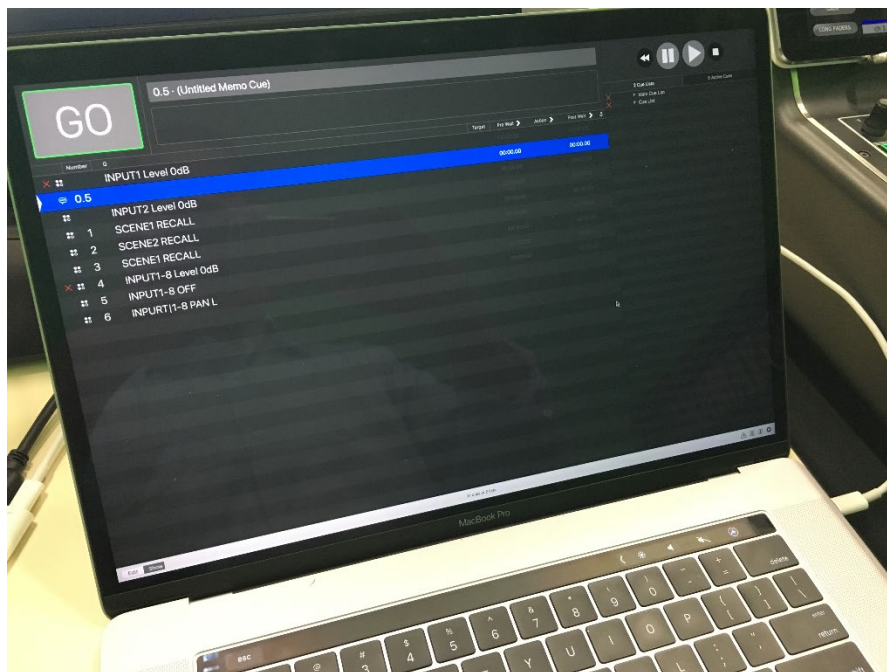
QLab のスクリプトキューを使用して制御可能な機能は以下の通りです。

- ・ シーンのリコール
- ・ 各入力チャンネルのオン/オフ
- ・ 各入力チャンネルのパン
- ・ 各入力チャンネルのフェーダーレベル

### 3. 使用機材

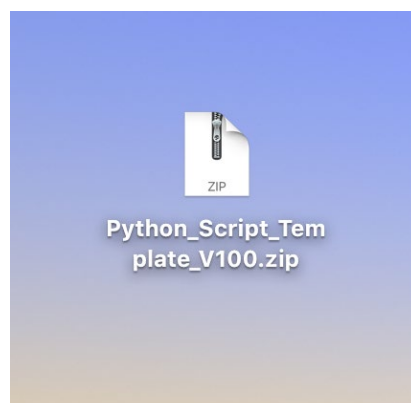
このガイドでは以下の機材を使用します。

- ・ Yamaha CL3 (Version 3.11以降)およびYamaha TF (Version2.00以降)
- ・ Mac (OSX 10.10[Yosemite]以降)上で動作するFigure 53社 QLab 4 (有償のAudioまたはBundleエディションが必要です)



【Figure 53 社 QLab 4 (Audio エディション)】

- ・ スクリプトテンプレート



【スクリプトテンプレート (zip ファイル)】

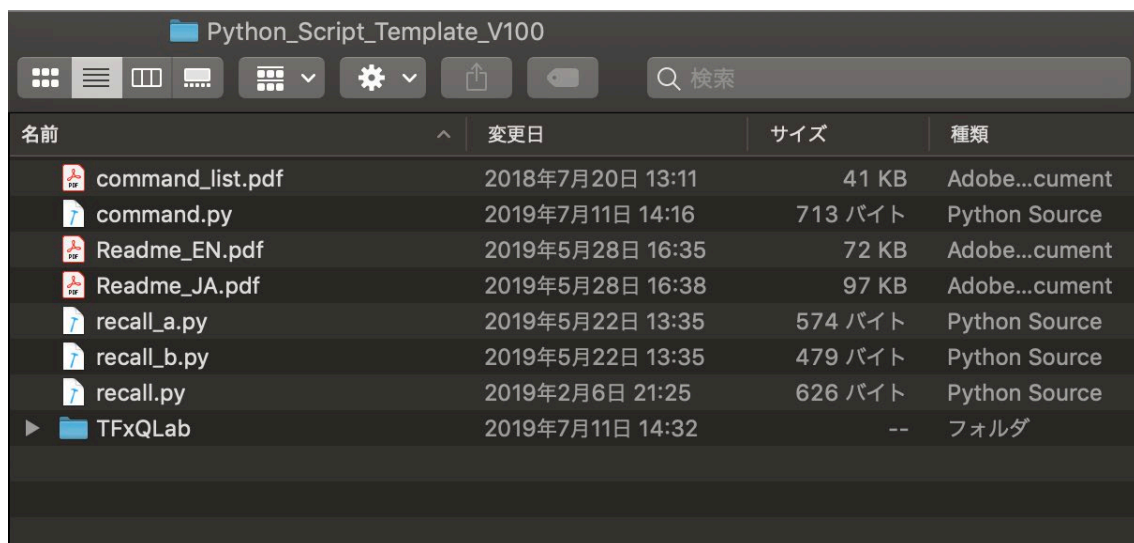
## 4. スクリプトテンプレートのダウンロード

スクリプトテンプレートは以下、CL/QL シリーズのウェブページの「ダウンロード」からダウンロードいただけます。

[https://jp.yamaha.com/files/download/other\\_assets/0/1266290/Python\\_Script\\_Template\\_V100.zip](https://jp.yamaha.com/files/download/other_assets/0/1266290/Python_Script_Template_V100.zip)

Data Sheet (CL Series)	—
Data Sheet (Rio3224-D, Rio1608-D, Ri8-D, Ro8-D)	📄 [1.8MB]
Data Sheet (RMio64-D)	—
Data Sheet (RSio64-D)	—
EQ Presets for In-ear Monitors (CL Series / QL Series)	—
MIDI Paramete Change List (CL/QL)	—
Python Script Template V1.0.0	📄 [894KB]
Rio3224-D, Rio1608-D Dimensions	—
Rio3224-D2, Rio1608-D2 Dimensions	—
RSio64-D Dimensions	—
USBフラッシュメモリー CL/QL/M7CL/LS9 動作確認リスト	📄 [73KB]
ヤマハDante機器と他社AES67機器の接続ガイド	📄 [556KB]
仕様書 (LA1L)	📄 [171KB]

ダウンロードしたファイルを展開すると、以下のようなファイルが見つかります。展開したフォルダはデスクトップに置いてください。



【展開後のスクリプトテンプレート】

## Python\_Script\_Template\_V100.zip に含まれるファイル

- ・ command\_list.pdf – リモートコントロール可能なコマンドの一覧
- ・ command.py – コマンドを扱うためのPythonスクリプト
- ・ Readme\_EN.pdf/Readme\_JA.pdf – 使用方法について書かれたreadmeファイル
- ・ recall\_a.py/recall\_b.py – シーンリコール用スクリプトファイル(TFシリーズ用)
- ・ recall.py – シーンリコール用スクリプトファイル(CL/QLシリーズ用)
- ・ TFXQLab folder – TF用サンプルパッケージ(QLabファイル、TFコンソールファイル、Pythonスクリプトファイル等含む)

## 5. コンソールと Mac の接続

スクリプトテンプレートファイルをダウンロードし展開したら、次にコンソールと Mac を LAN ケーブルで接続します。スイッチを使用しても、またお互いのデバイスを直接接続しても構いません(Dante のポートではなく、ミキサーの LAN ポートに接続してください)。



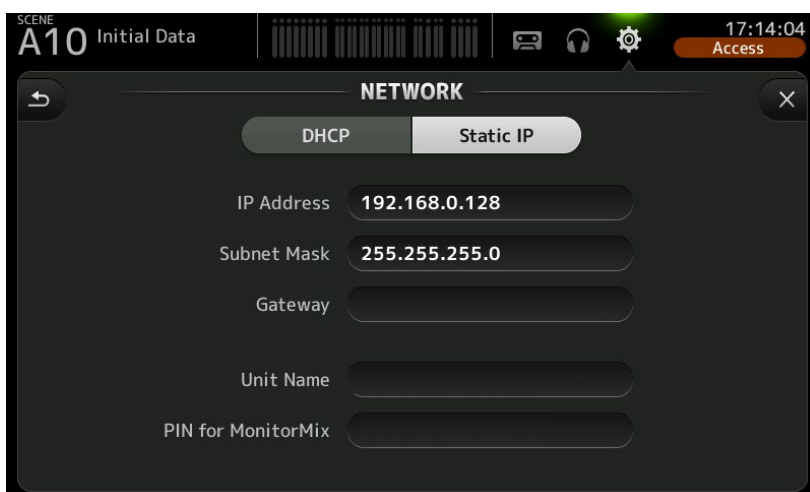
【LAN ケーブル経由でのコンソールと Mac を接続】

## 6. IP アドレスの設定

次にコンソールと Mac のそれぞれの IP アドレスを設定します。CL/QL シリーズは NETWORK 画面の FOR MIXER CONTROL にアクセスし、IP アドレスに 192.168.0.128、サブネットマスクに 255.255.255.0 を選択します。TF シリーズは設定画面から NETWORK を選択し、Static IP 画面に上記 IP アドレスとサブネットマスクを入力してください。



【NETWORK – FOR MIXER CONTROL(CL/QL)】



【NETWORK – Static IP(TF)】

MacではIPアドレスに同じネットワークアドレスのアドレスとサブネットマスクを設定します。設定-ネットワークから以下の項目を選択、入力してください。

- ・ IPv4の設定                    手入力
- ・ IPアドレス                    192.168.0.100
- ・ サブネットマスク            255.255.255.0



【設定 - ネットワーク】

## 7. スクリプトテンプレートの設定

CL/QL の場合、次に展開したスクリプトファイルの設定を確認します。まずは「command.py」を開き、host(コンソール)のIPアドレスがコンソールに設定したアドレスになっているか確認してください。もし万が一設定した以外のIPアドレスが設定されている場合、アドレスを修正し、ファイルを保存します。

```

3 # "python command.py ssrecall_ex MIXER:Lib/Scene
4 # It will recall scene #001 of the target console
5
6 import sys
7 import socket
8
9 # Host is console's IP
10 host = "192.168.0.128"
11 # Port must be 49280
12 port = 49280
13
14 # create command
15 args = sys.argv
16 args.pop(0)
17 command = ''
18 for tmp in args:
19     if command != '':

```

【command.py】

同じように「recall.py」の中の host のアドレスも同じコンソールのアドレスになっているか確認してください。

```

3 # "python recall.py 1"
4 # It will recall scene #001 of the target console
5
6 import sys
7 import socket
8
9 # Host is console's IP
10 host = "192.168.0.128"
11 # Port must be 49280
12 port = 49280
13
14 args = sys.argv
15 no = int(args[1])
16
17 # connect socket
18 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
19 s.settimeout(5)

```

【recall.py】

TF シリーズの場合、「TFxQLab」フォルダ内の「recalla0.py」, 「recallb1.py」および「MultiCh.py」をそれぞれ開き、同じく host の IP アドレスがコンソールに設定したアドレスになっているか確認してください。

## 8. QLab のプログラミング(CL/QL シリーズの場合)

QLab から制御可能な CL/QL シリーズの機能やパラメーターは以下の通りです。

- ・ シーンリコール
- ・ 各入力チャンネルのオンオフ
- ・ 各入力チャンネルのパンおよびフェーダーレベル

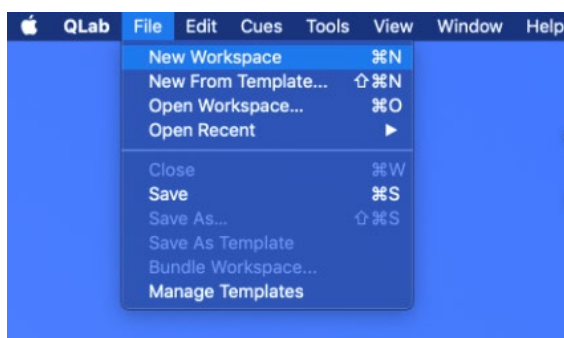
コンソールのコントロールには QLab の「スクリプトキュー」を使用します。

QLab のスクリプトキューを実行することにより、QLab 上で AppleScript が実行されます。この例では QLab が AppleScript を使用して python スクリプトを走らせ、その python スクリプトが制御コマンドをコンソールに送ります。

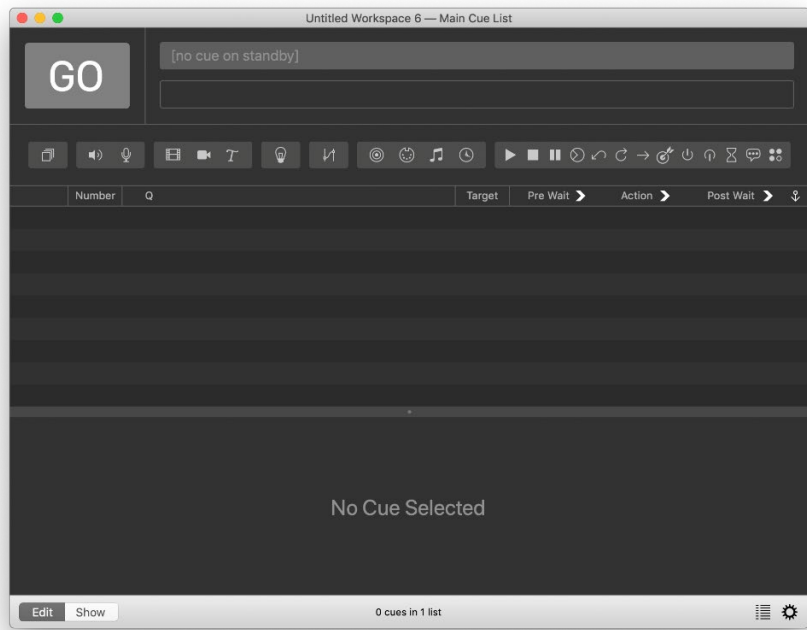
### QLab 4 スクリプトキュー



QLab の File メニューから New Workspace を選択するとブランクのワークスペースが開きます。

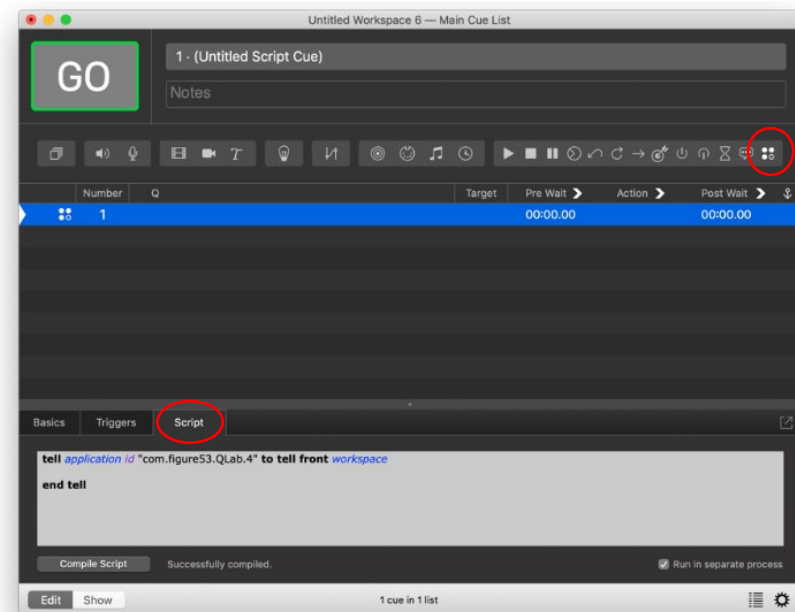


【QLab File メニュー】



【QLab ブランクワークスペース】

☰ ボタンをクリックすると空白のスクリプトキューが作成されます。画面下の「Script」タブをクリックすると、AppleScript をプログラムするテキストフィールドが表示されます。



【QLab スクリプトキュー】

## 8-1. シーンのリコール

フィールド内にはデフォルトで以下のテキストがあらかじめ入力されています。

```
tell application id "com.figure53.QLab.4" to tell front workspace
end tell
```

上記 2 行の間に、以下の 1 行を入力します。

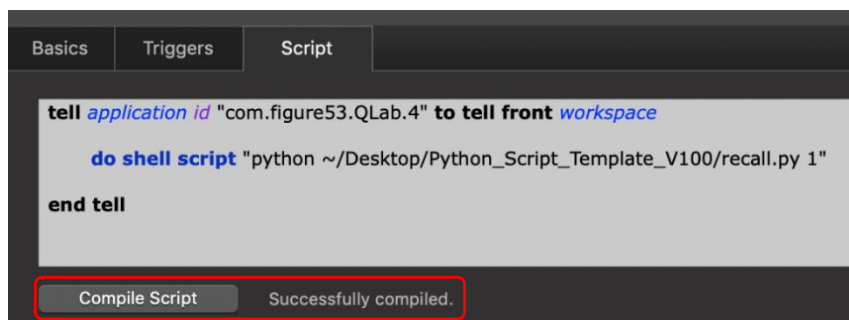
```
do shell script "python ~/Desktop/Python_Script_Template_V100/recall.py 1"
```

```
tell application id "com.figure53.QLab.4" to tell front workspace
  do shell script "python ~/Desktop/Python_Script_Template_V100/recall.py 1"
end tell
```

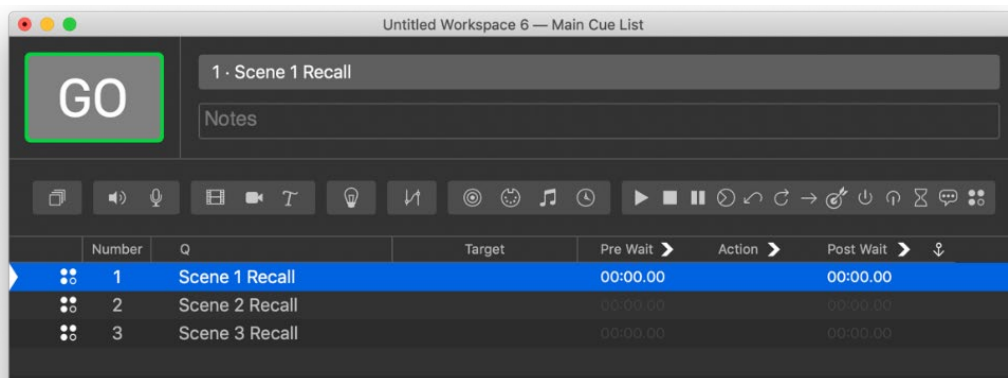
行末の数字(この場合は 1)がリコールするシーン番号となりますので、もしシーン 8 をリコールしたい場合は以下のように入力します。

```
do shell script "python ~/Desktop/Python_Script_Template_V100/recall.py 8"
```

入力が済んだら、下の「Compile Script」ボタンを押して、スクリプトが問題なくコンパイルできるか確認してください。



作成した各スクリプトキューには、スクリプトの内容を覚えやすいように名前を付けます(「Scene 1 Recall」など)。



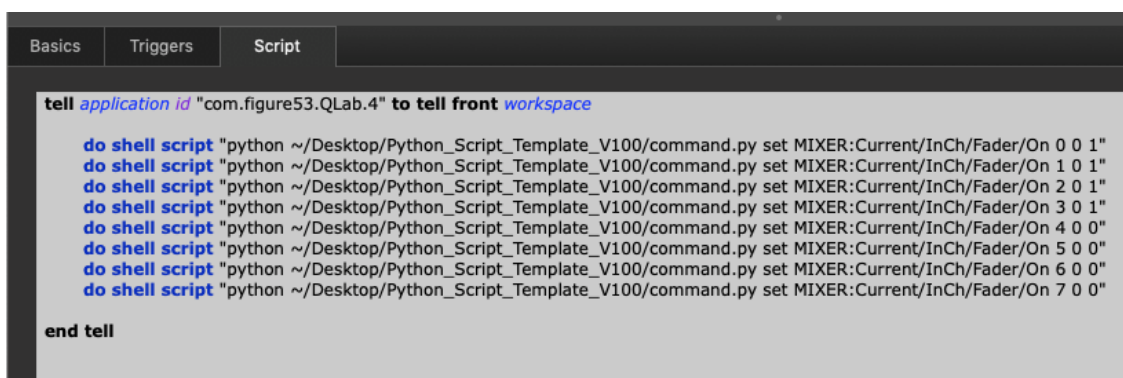
各列をクリックし、「GO」をクリックするかスペースバーを押すと、コンソールの該当のシーンがリコールされます。

## 8-2. 各入力チャンネルのオンオフ、パン/フェーダーレベルの設定

シーンリコールに加え、各入力チャンネルのオン・オフ、パンレベル、およびフェーダーレベルを一斉に、または個別に設定することが可能です。

まず、入力チャンネル 1-4 のチャンネルをオンにし、5-8 をオフにするスクリプトを実行します。新規スクリプトキューを追加し、以下のテキストを入力(コピーペースト)してコンパイルします。

```
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/On 0 0 1"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/On 1 0 1"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/On 2 0 1"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/On 3 0 1"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/On 4 0 0"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/On 5 0 0"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/On 6 0 0"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/On 7 0 0"
```



コンパイルが済んだらスクリプトを実行し、以下のようにチャンネル 1-8 のオンオフが切り替わることを確認してください。



【チャンネルオンオフのリモートコントロール】

インプットチャンネル 1-8 のパンを設定するにはスクリプトキューを新規作成し、以下の行を入力してコンパイルします。最小値の-63 が左端、0 が中央、最大値の 63 が右端になります。

```
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/ToSt/Pan 0 0 -63"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/ToSt/Pan 1 0 -45"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/ToSt/Pan 2 0 -30"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/ToSt/Pan 3 0 -10"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/ToSt/Pan 4 0 10"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/ToSt/Pan 5 0 30"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/ToSt/Pan 6 0 45"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/ToSt/Pan 7 0 63"
```

インプットチャンネル 1-8 のフェーダーレベルを設定するにはスクリプトキューを新規作成し、以下の行を入力してコンパイルしてください。

```
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 0 0 0"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 1 0 -1000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 2 0 -2000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 3 0 -3000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 4 0 -5000"
```

```
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 5 0 -10000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 6 0 -12000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 7 0 -23285"
```

```
asics Triggers Script
tell application id "com.figure53.QLab.4" to tell front workspace
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 0 0 0"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 1 0 -1000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 2 0 -5000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 3 0 -10000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 4 0 -12000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 5 0 -15000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 6 0 -20000"
do shell script "python ~/Desktop/Python_Script_Template_V100/command.py set MIXER:Current/InCh/Fader/Level 7 0 -23285"
end tell
```

以下はフェーダーレベルとフェーダー値の関係を簡単に表した表です。

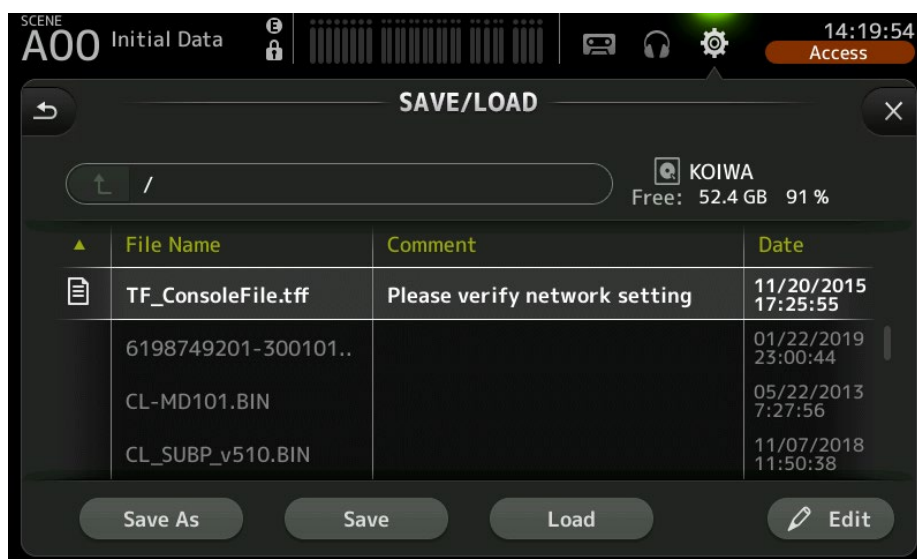
フェーダーレベル	フェーダー設定値
+10dB	1,000 [最大値]
+1dB	100
0dB	0
-10dB	-1,000
-20dB	-2,000
-30dB	-3,000
-50dB	-5,000
-99dB (-100dB はない)	-10,000
-120dB	-12,000
$-\infty$ dB	-32,768[最小値] - -23,285

## 9. QLab のプログラミング (TF シリーズの場合)

今回、TF シリーズユーザー向けに、「TFxQLab」フォルダの中に以下のファイルを用意しました。「TFxQLab」フォルダはダウンロードし展開したフォルダから取り出し、デスクトップに移動させておいてください。

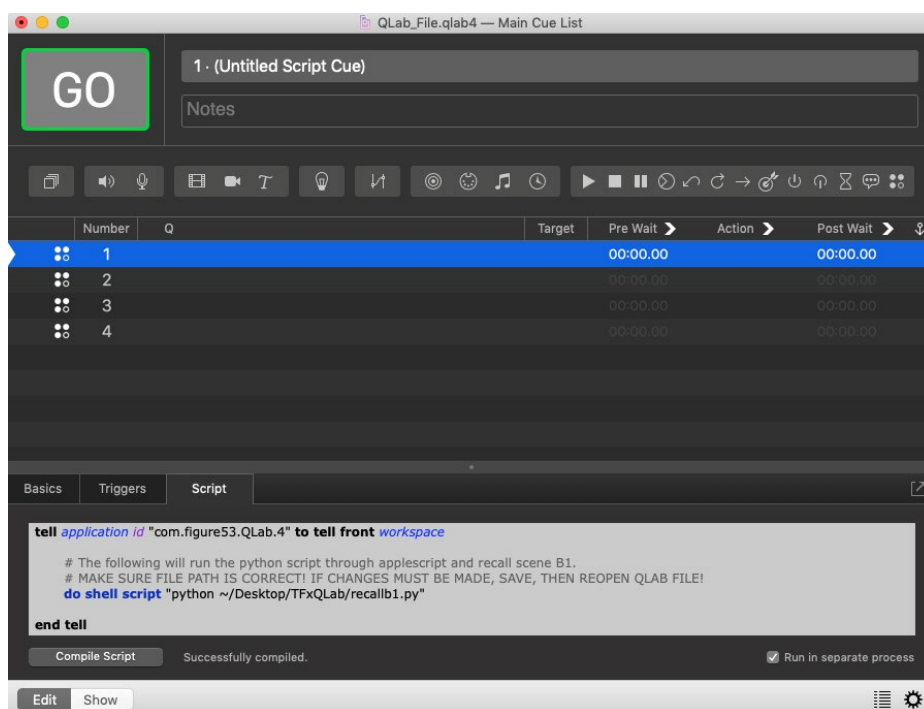
- ・ TF用コンソールファイル(TF\_ConsoleFile.tff)
- ・ QLab4用ワークスペースファイル(QLab\_File.qlab4)
- ・ recalla0.py
- ・ recallb1.py
- ・ MultiCh.py

まずは TF にコンソールデータをロードします。



【コンソールファイルのロード】

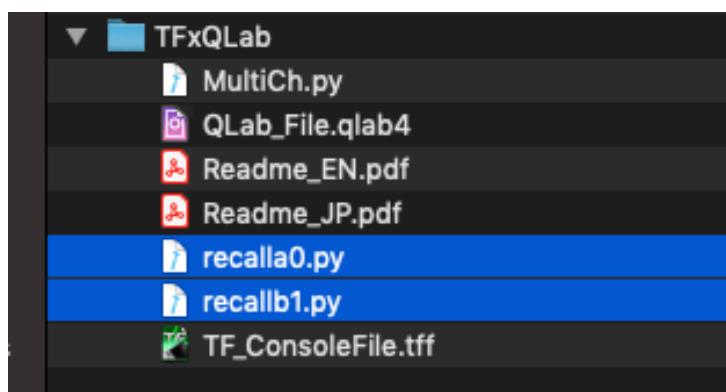
次に QLab を起動し、“QLab\_File.qlab4”ファイルを開きます。



【QLab\_File.qlab4】

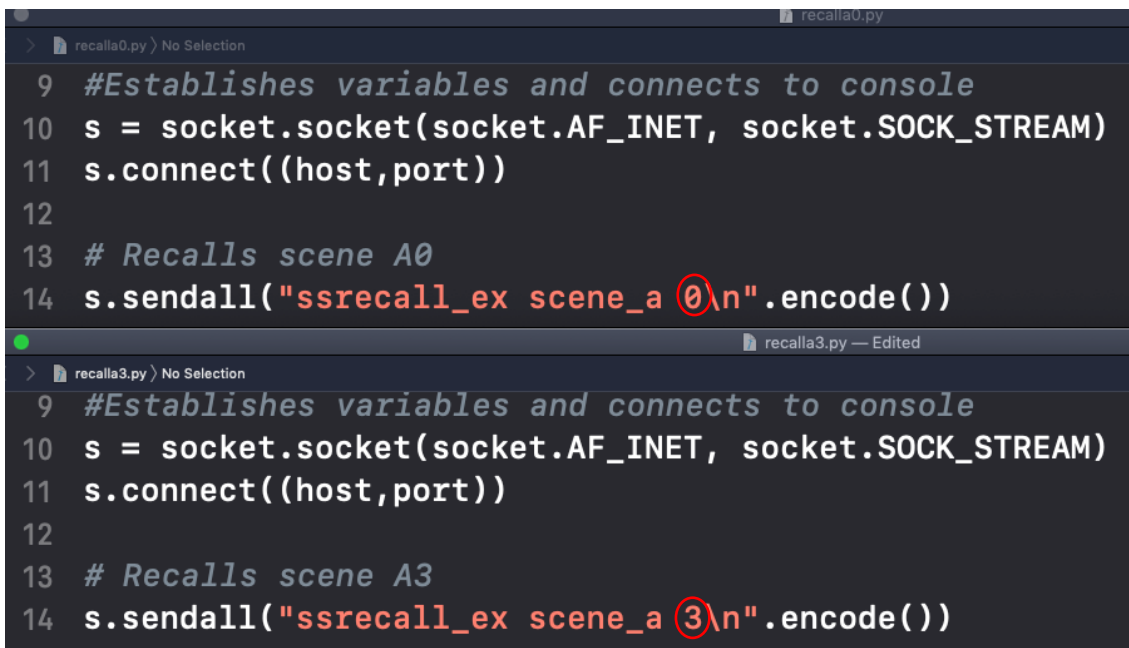
## 9-1. シーンのリコール

TF シリーズでは A/B の 2 グループのシーンがありますが、2 つのサンプルスクリプトはそれぞれ A0 と B1 をリコールするためのものです。



【TFxQLab フォルダ】

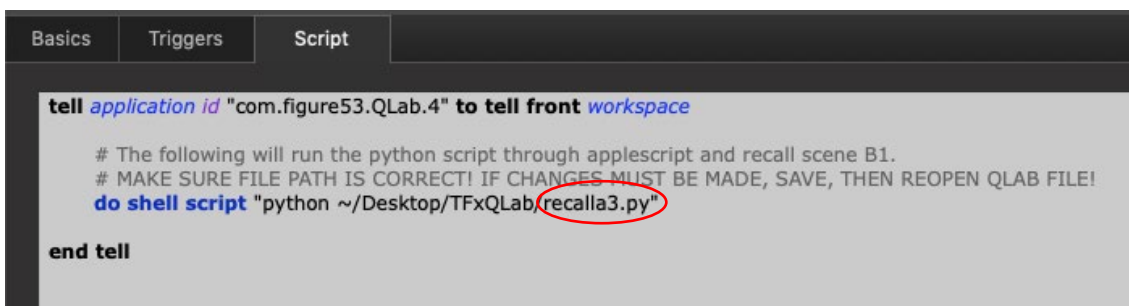
シーンの B1 をリコールするには QLab 上で「recallb1.py」を実行し、A0 をリコールするには「recalla0.py」を実行します。もし「recalla0.py」を編集して別のシーンをリコールしたい場合は「<sup>ひきすう</sup>引数」(関数の実行に必要な変数)を 0 から他の値に変更するだけです。変更が済んだら「recalla3.py」など別のファイル名で保存してください。



```
9 #Establishes variables and connects to console
10 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
11 s.connect((host,port))
12
13 # Recalls scene A0
14 s.sendall("ssrecall_ex scene_a 0\n".encode())

recalla3.py — Edited
9 #Establishes variables and connects to console
10 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
11 s.connect((host,port))
12
13 # Recalls scene A3
14 s.sendall("ssrecall_ex scene_a 3\n".encode())
```

その後、QLab4 のスクリプトキューにアクセスし、実行するスクリプトファイル名を変更します。



```
Basics Triggers Script
tell application id "com.figure53.QLab.4" to tell front workspace
    # The following will run the python script through applescript and recall scene B1.
    # MAKE SURE FILE PATH IS CORRECT! IF CHANGES MUST BE MADE, SAVE, THEN REOPEN QLAB FILE!
    do shell script "python ~/Desktop/TFxQLab/recalla3.py"
end tell
```

```
do shell script "python ~/Desktop/TFxQLab/recalla3.py"
```

## 9-2. 各入力チャンネルのオンオフ、パン/フェーダーレベルの設定

この例では、「MultiCh.py」を編集し、複数の入力チャンネルのオンオフとフェーダーレベルを一度に設定します。

```
18 # turn channel 1 off
19 s.sendall("set MIXER:Current/InCh/Fader/On 0 0 0\n".encode())
20
21 # turn channel 2 off
22 s.sendall("set MIXER:Current/InCh/Fader/On 1 0 0\n".encode())
23
24 # set channel 3 fader to 0
25 s.sendall("set MIXER:Current/InCh/Fader/Level 2 0 0\n".encode())
26
27 # set channel 4 fader to 0
28 s.sendall("set MIXER:Current/InCh/Fader/Level 3 0 0\n".encode())
29
30 # set channel 5 fader to +5
31 s.sendall("set MIXER:Current/InCh/Fader/Level 4 0 500\n".encode())
32
33 # set channel 6 fader to -10
34 s.sendall("set MIXER:Current/InCh/Fader/Level 5 0 -1000\n".encode())
35
```

【MultiCh.py】

入力チャンネルのオンオフを設定するには以下の赤字と青字の数値を変更します。

```
s.sendall("set MIXER:Current/Inch/Fader/On 0 0 0\n".encode())
```

赤はチャンネルナンバー(0=入力1、1=入力2...)

青はオンオフ(1=オン、0=オフ)

同様に、パン/フェーダーレベルを設定するには以下を編集します。

```
s.sendall("set MIXER:Current/Inch/ToSt/Pan 0 0 0\n".encode())
```

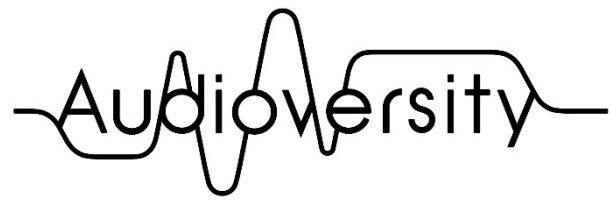
赤はチャンネルナンバー(0=入力1、1=入力2...)

青はパンレベル(-63=L63, 0=Center, 63=R63)

```
s.sendall("set MIXER:Current/Inch/Fader/Level 0 0 0\n".encode())
```

赤はチャンネルナンバー(0=入力1、1=入力2...)

青はフェーダーレベル(-32,768=-Infinity dB, 1000=+10.00 dB)



by Yamaha Pro Audio