

For practical reasons we want the **height** of the Mixer window in *MIDI View* to be always the same. If we have selected a one-staff instrument and next we select an organ with three staves the first thing we notice is that the height of the strips decreases because of the increasing number of *Mute Voice* rows. In the Details Area we notice a vertical displacement of the *Sound select*, *Volume*, *Pan* and *Track color* rectangles. The height of *Play part only*, *Drumset*, *Reverb* and *Chorus* does not change.

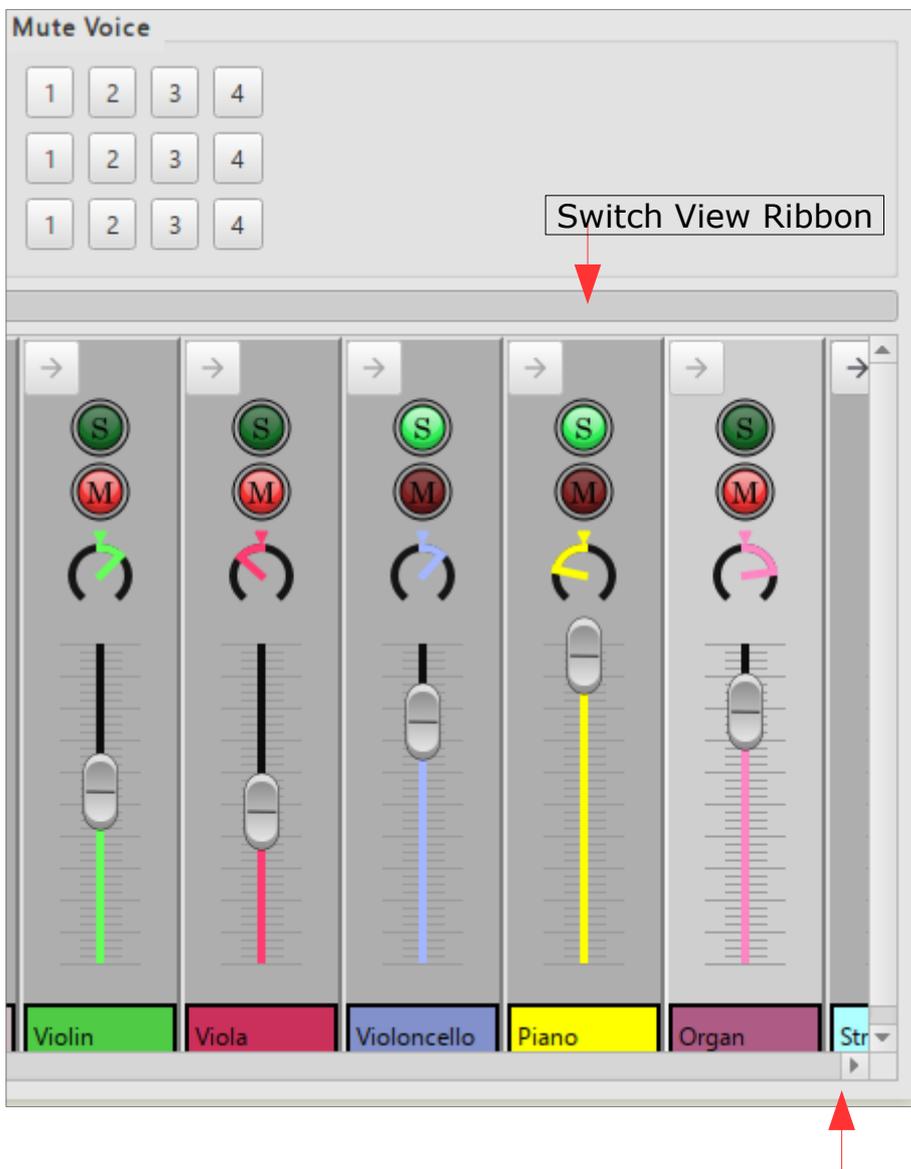
When we increase the **width** of the Mixer window because we want to display more strips most control elements in the Details Area will shift to the right.

The automation of the controls in the Details Area is restricted to instruments having a maximum of three *Mute Voice* rows. That could be an organ or piano with three staves or any instrument with two linked staves.

In the rather rare cases of four or more *Mute Voice* rows these controls will still need manual mouse movements.

For the *Strips only View* there is no restriction. There the height of the strips is always the same and *Arrow*, *Solo*, *Mute*, *Pan* and *Slider* are controlled irrespective of the number of *Mute Voice* rows.

Defined State: setting the height of the Mixer window



Create an empty score containing 10 staves. The ninth staff is Piano, the tenth is Organ. Display the window in *MIDI View* and select the *Organ strip*. Set the window height so that the vertical scroll bar is visible.

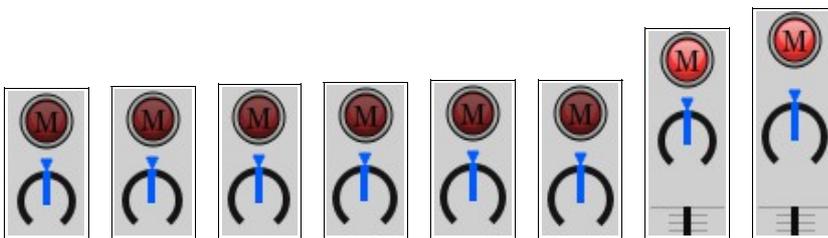
Start Window Spy and the tool PixelMousing. Go with the mouse to the window border (the point of the red arrow) and steer the mouse downward using PixelMousing. When the cursor changes shape enable *drag*.

When the vertical scrollbar disappears disable *drag*. In Window Spy we see for the test screen:

Active Window Position:				
	x: 498	y: 160	w: 941	h: 774
Client:	x: 0	y: 0	w: 925	h: 735

Excellent! We have found the value of our first Mixer variable. If the Mixer macros are *part of your AHK Kit* enter in *coordinates.ahk* the value found by you. If you use the file as a stand-alone group of macros the coordinates are part of the auto-execute section of the file. See *General remarks* on page 16 of this document. Anyway it concerns this variable: **Mixer_Height := 774**

So when the Mixer is displayed for the first time its height will be **774**. Next click somewhere in the *Switch View ribbon* to get to the *Strips only View*. The strips are now much too high. We don't need them that way. With Window Spy active we reduce the height - dragging by PixelMousing - until the vertical scrollbar appears. Now the height obviously is too small. We have to increase it again until the vertical scrollbar just disappears. But personally I prefer a slightly higher window.



Look at the *Pan* dial. When you further increase the height the diameter of the dial increases as well and the distance between *Pan* and *Mute* too. Also the top of the slider ruler goes down. When you reach the point where the top of the ruler doesn't go further down the height of the *Strips only View* is IMO ergonomically optimal. So now we have found the value of **StripsOnlyHeight**.

Active Window Position:				
	x: 498	y: 370	w: 924	h: 507
Client:	x: 0	y: 0	w: 908	h: 468

StripsOnlyHeight := 507

When we press **F10 + X** the **Mixer_Height** will toggle between **StripsOnlyHeight** and **MidiHeight**. So in *coordinates.ahk* (or the auto-exec section) we enter the value already found for **Mixer_Height** a second time but now for this variable and get **MidiHeight := 774**.

Defined State: Verification of the view

Without any further precaution **F10 + X** could easily fail. The failure is caused by the fact that in *MIDI View* the height of the *Switch View ribbon* depends on the number of *Mute Voice* rows. We have to account for the possibility that we switch the *View* while we have an instrument selected with one, two or three staves. Also in some situations it is still possible that after **F10 + F11** we get the size of the *MIDI View* but without the MIDI controls, resulting in a very high strip. Therefore we have to check the status of the *MIDI View*. When needed the macros will search for this image  in front of its rectangle and report at which height it is found. The found height will differ dependent on having an instrument selected with one, two or three staves.

ImageSearch, , Instr_Status, Sound_X1, Sound_Y1, Sound_X2, Sound_Y2, MXR_Sound.png

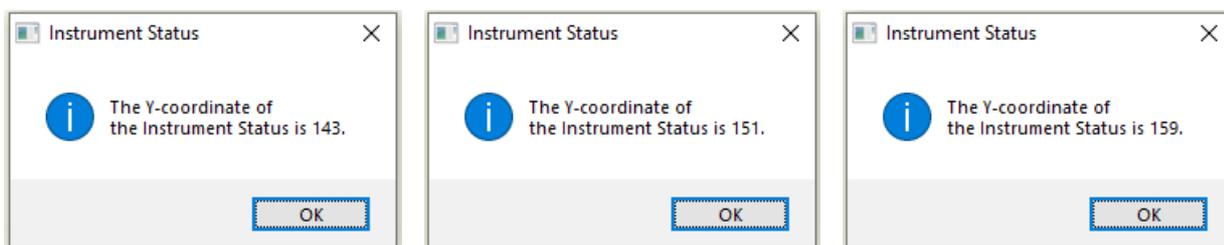
First create this image using the Snipping Tool. Put **MXR_Sound.png** in your working directory. Next determine the coordinates of the search surface with PixelMousing. So make the height of the window **MidiHeight**. We see that the horizontal position of this image is always the same.

Sound_X1 and **Sound_Y1** are the coordinates of the upper-left corner of the search area. So determine these numbers with a *one-staff-instrument selected*. **Sound_X2** and **Sound_Y2** are the coordinates of the lower-right corner of the search area. So determine these numbers with an *organ selected*. For the test screen these values were found:

Sound_X1 :=8 Sound_Y1 := 143 Sound_X2 :=57 Sound_Y2 := 178

After the **ImageSearch** the found height of the upper-left corner of **Sound:** is stored in the variable **Instr_Status**.

To determine the number of **Instr_Status** the Mixer macros include a small utility which you can invoke with the hotkey **Z + 6**. You have to do it thrice, for *Normal* instrument, *Piano* and *Organ*. The numbers will increase in this order.



In *coordinates.ahk* (or the *autoexec* section) you enter the found values.
Instr_Status_Normal := 143 Instr_Status_Piano := 151
Instr_Status_Organ := 159

If in the *Show/Hide* command **F10 + F11** the Mixer has the size of the *MIDI View* but **Sound:** is not found the mouse will click the *Expand* point in the *Switch View ribbon*. This point always has the same Y-coordinate irrespective of the height of the strip.

In the *Switch View* command **F10 + X** the mouse must also click somewhere in the *Switch View ribbon*. When the image is not found the mouse clicks on the *Expand* point. When the image is found the mouse clicks on the *Collapse* point. The height of this point is determined by the value of **Instr_Status**.

We'll return to the *Collapse* and *Expand* points below in the description of all specific hotspots. See page 7.

Defined State: setting the width of the Mixer window

After pressing **F10 + F11** for the first time the Mixer will be displayed in its default width of 10 strips. How do we get the values of this Defined Width?

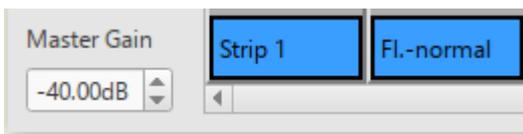
Create an empty score containing 40 instruments. Use the template Symphony Orchestra which gives you immediately 25 staves. Using Pixel-Mousing while dragging a vertical window border we set the width of the Mixer so that it contains *exactly* 10 strips. The picture shows the optimal layout.



If we scroll by clicking **R** the result must be that we see strip 11 thru 20 while the view remains static, without the slightest shift. Idem if we click **L**.

It must look as if the numbers in Strip 1,2,3,4,5,6,7,8,9 suddenly get a '1' before their number and become 11,12,13 etc.

Notice that the left edge of the strip - in this theme - is whitish and the right edge is a darker grey. Window Spy tells us the width.

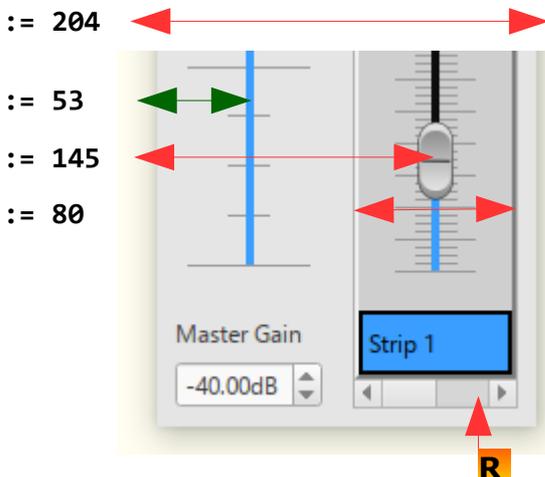


Strip + neighbour OK

Active Window Position:			
x:	321	y:	149
w:	924	h:	774
Client:	x: 0	y:	0
	w:	908	h: 735

Mixer_Width := 924

is the default value for 10 strips.



Next we set the mixer width in *Strips only View* so that it contains *exactly* one strip. If we scroll by clicking **R** the view must remain static. Again WindowSpy will tell us the width. The window also includes narrow shadow areas seemingly outside the window. Where the window horizontally begins and ends the mouse will change shape, from 'pointer' to 'separator'.

Notice: the first pixel is pixel zero.

MasterGainAndStripOne_X := 204

The width difference between a Mixer with 10 strips and 1 strip is 9 strips!
So it's rather easy to check **StripWidth**.

Mixer_Width (924) minus MasterGainAndStripOne_X (204) = 720 divided by 9.
Check if **StripWidth := 80** is correct and enter your values.

The second red line in the previous picture refers to the distance between the left border of *Master Gain* and a point in the **middle** of the fader. This is **MidFirstStrip_X := 145** and it provides the X-coordinate for control elements as *Solo* and *Mute* in the first strip. The third red line is **StripWidth := 80**. The green line refers to the distance of the Master Gain. **MasterGain_X := 53**
MidFirstStrip_X and **StripWidth** will return in the description of all specific hotspots of the strip control elements.

Now we have all the data needed to set the Mixer width for any amount of strips. This can be handy in a big score if we want to focus on say a Wind or String section. But let us first give the Mixer window a nice position on the screen.

Positioning the Mixer window on the screen: Mixer_Y

Irrespective of the width of the Mixer window we can have our preferences about how high (or low) the window must be positioned on the screen. This concerns the Y-coordinate of the upper-left corner of the window relative to the screen.

For instance we could prefer that the window in *Strips only View* is displayed at the same screen height as in *MIDI View* so that the strips will be displayed where a moment ago *Reverb* and *Chorus* were visible. Or we could prefer that *Strips only View* must be positioned lower.

Comparable with our discussion of **Mixer_Height** we have three forms of **Mixer_Y**. We have the default Y-position when we display the Mixer for the first time: **Mixer_Y := 160**. That is the same as its position in *MIDI View* **MidiMixer_Y := 160**. My favorite is a lower position: in *Strips only View* **StripsOnlyMixer_Y := 427**. With these value the strips occupy exactly the same surface on the screen in both views. Switching to *MIDI View* looks then more like adding some extra useful space to an already existing window. Notice that in this case **StripsOnlyMixer_Y minus MidiMixer_Y (427 - 160)** is the same as **Midi_Height minus StripsOnlyHeight (774 - 507)** being (267).

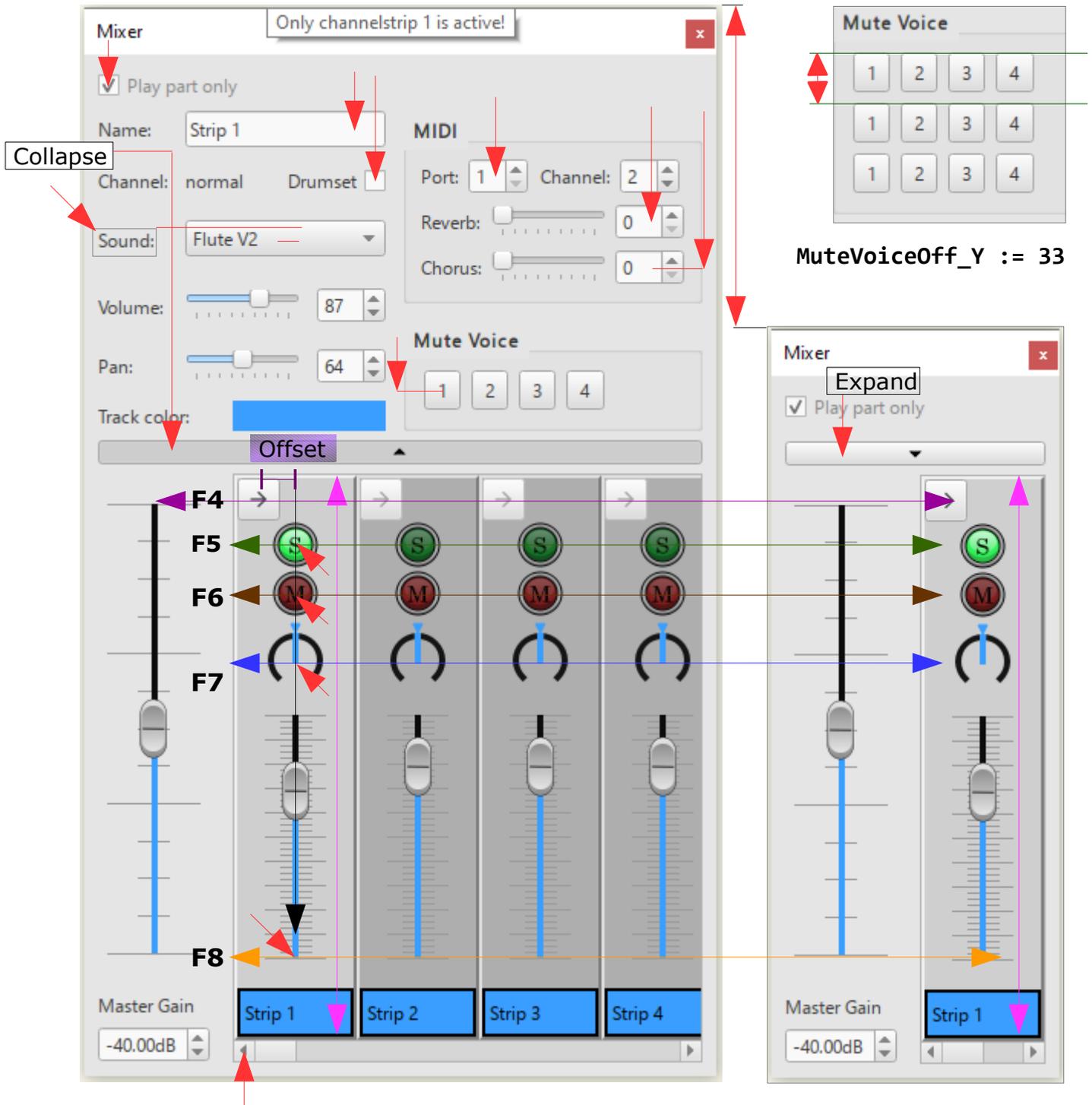
Determining a horizontal coordinate is superfluous because the middle of the Mixer window will be positioned in the middle of the screen. AutoHotkey can detect the screen width. It has an in-built variable **A_ScreenWidth**.

This means that **Mixer_X** can be simply derived thus:

Mixer_X := (A_ScreenWidth - Mixer_Width) divided by 2.

Determining the Y-coordinate of the control element hotspots and already a few X-coordinates of elements which don't shift

NB: the smallest *MIDI View* will display 4 strips. But to have room for one strip (in *Strips only View*) on this page the pics show a cut 4th strip in *MIDI View*.



The X-coordinate of the *Collapse* and *Expand* points is the same. Choose it so that the point in *MIDI View* lies between *Name* and its rectangle.

$MxrExpandCollapse_X := 60$ $MxrExpand_Y := 80$ and - for a one staff instrument $MxrCollapseMidi_Y := 291$. If *Instr_Status* is *Piano* or *Organ* once resp. twice *MuteVoiceOff_Y* will be added.

Determine now

```
PlayPartOnly_X := 23 *           PlayPartOnly_Y := 49
PartName_X := 195 *             PartName_Y := 80 ("Strip 1")
MxrSoundSel_X := 194 *
LeftScrollTriangle_X := 110 *   ScrollHeightMidi_Y := 749
* Determine these X-coordinates in a window containing 4 strips
```

```
MxrSoundSelOff_Y := 7
```

This is the distance between the red lines in the rectangle *Flute V2*. See pic. The highest line gets its Y-coordinate from **Instr_Status** so the offset depends on the dimensions of your image **Sound:** The offset ensures that the Sound select rectangle is clicked halfway its height.

```
Drumset_Y := 113
MidiPortChannel_Y := 110       Port and Channel have the same height
Reverb_Y := 140
Chorus_Y := 170
MidiMuteVoice_Y := 252        For Instr_Status_Normal
TrackColor_Y := 268          For Instr_Status_Normal
Volume
Pan                            These two don't need to be set.
                                They are controlled in Strips only View
```

All control elements in the *Strips only View* exist also in the *MIDI View*. For **Instr_Status_Normal** - instruments with only one staff - they have the same difference in height between the two views. Determine their height in the *MIDI View* for **Instr_Status_Normal**

```
MidiViewShow_Hide_Ch_Y := 322   Arrow
MidiViewSolo_Y := 350           Solo
MidiViewMute_Y := 382          Mute
MidiViewPan_Y := 440           Pan
MidiViewSlider_Y := 688        Slider
MidiViewMaster Gain_Y is superfluous. It is MidiViewShow_Hide_Ch_Y.
```

Switch now to *Strips only View* - with still **Instr_Status_Normal** - and position the mouse in the centre of *Mute*. Let us call the Y-coordinate of this spot **StripsOnlyMute_Y**.

```
Ctrl_Element_Diff_Y = MidiViewMute_Y minus StripsOnlyMute_Y.
Ctrl_Element_Diff_Y := 211
```

In *Strips only View* the macros will dutifully subtract **Ctrl_Element_Diff_Y** from the coordinates of *Arrow* (is also *MasterGain*), *Solo*, *Mute* and *Pan*. In *MIDI View* however when the status is **Instr_Status_Piano** or **Instr_Status_Organ** once respectively twice **MuteVoiceOff_Y** will be added to **MidiViewSolo_Y** and companions.

MidiViewSlider_Y and **ScrollHeightMidi_Y** follow another approach. Their hotspots are very low at a fixed distance from the lower border of the window and are not displaced by rows of *Mute Voice*.

In *Strips only View* their height is determined by the height difference between the two *Views*. It furthers readability to use a derived variable:

MxrHeightDiff := **MidiHeight** - **StripsOnlyHeight**. So we can get well readable formulas like:

ScrollHeightStripsOnly_Y := **ScrollHeightMidi_Y** - **MxrHeightDiff**

Determining the X-coordinate of the control elements hotspots

MasterGain_X := 53 **MidFirstStrip_X** := 145 and **StripWidth** := 80

These we have found already. With the last two we can easily calculate the X-coordinate of almost all control elements of the *Strips only View*.

Just by adding **StripWidths** to **MidFirstStrip_X**.

The only exception is the arrow of *Show channels* which lies a bit to the left of **MidFirstStrip_X**. We have to determine this distance, marked by **Offset** in the big picture displayed above. The variable is **Show_Hide_Off_X** := 21 .

Subtracted from **MidFirstStrip_X** this gives us the X of *Show channels*.

The values are valid for both the *MIDI View* and the *Strips only View*. For 2 strips we add 1 **StripWidth**, for 3 strips 2 and so on.

The computer loves this type of simple calculations: for instance the middle of strip 10 will be **MidFirstStrip_X** + (9 x **StripWidth**) Etc.

Scaling X-coordinates of control elements in the Details Area

The X-position of *Drumset* and *Track color* and of all MIDI control rectangles on the right half of the Mixer window depends on the amount of strips. As mentioned the minimum width in *MIDI View* is 4 strips. Most control elements have a separate position when there are 5 strips. When there are 6 strips or more the displacements of all control elements are regular. The X-coordinates increase then with 1/2, 3/4 or 1 stripwidth.

Drumset and Track color

We use the same X-coordinate for the rectangles of *Drumset* and *Track color*.

For a Mixer with *exactly* 4 strips: determine **Drumset_4_X** := 198

For a Mixer with *exactly* 5 strips: determine **Drumset_5_X** := 220

For a Mixer with *exactly* 6 strips: determine **Drumset_6_22_X** := 289

With each extra strip the position increases by 1/2 **StripWidth**.

Reverb and Chorus

The rectangles of *Reverb* and *Chorus* have the same X.

For a Mixer with *exactly* 4 strips: determine `Reverb_Chorus_4_X := 396`
With each extra strip the position increases by 1 StripWidth.

Midi Port

For a Mixer with *exactly* 4 strips: determine `MidiPort_4_X := 278`
For a Mixer with *exactly* 5 strips: determine `MidiPort_5_X := 300`
For a Mixer with *exactly* 6 strips: determine `MidiPort_6_22_X := 370`
With each extra strip the position increases by 1/2 StripWidth.

Midi Channel

For a Mixer with *exactly* 4 strips: determine `MidiChannel_4_X := 376`
For a Mixer with *exactly* 5 strips: determine `MidiChannel_5_X := 436`
For a Mixer with *exactly* 6 strips: determine `MidiChannel_6_22_X := 510`
With each extra strip the position increases by 3/4 StripWidth.

Mute Voice

Determine the coordinate for *Mute voice 1*.

For a Mixer with *exactly* 4 strips: determine `MidiMuteVoice_1_4_X := 242`
For a Mixer with *exactly* 5 strips: determine `MidiMuteVoice_5_X := 265`
For a Mixer with *exactly* 6 strips: determine `MidiMuteVoice_6_22_X := 334`
With each extra strip the X position increases by 3/4 StripWidth.

The X-coordinates of Mute Voice 2, 3 and 4 are calculated by adding an offset.



Measure the length of the red line and divide by 3 to find the offset.

`MidiMuteVoiceOffset_X := 31`

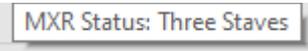
Congratulations! You have finished determining all values of the Mixer window!

But it could still be that you want to change the position of some tooltips appearing in several commands. In the macros these items are marked with (*change*?).

Explanation of Mixer commands

Control + F10 Toggle Mixer Status

After pressing **F10 + F11** to show the Mixer a tooltip appears at the top of the screen to indicate the actual status. This is especially important for *MIDI View*.

 or 

One Staff: Each selected strip is treated as having one staff. There will be no verification of **Instr_Status**. Selection of control elements will be fast. Selecting a piano or organ can result in mistakes in the *Track Area* and in the *Details Area*.

Three Staves. After selection of a new strip **Instr_Status** will be verified by clicking first the *Slider*. This adjusts several heights for correct selection of all control elements. This process is slower.

For instance with strip 1 selected - say a flute with one staff - you have just set *Solo* by pressing **F10 + F5**. Strip 9 is a piano with two staves. When you press **F10 + 9** the mouse first clicks the piano strip on the *Slider* hotspot and next clicks its *Solo* button.

For the fastest workflow only enable *Three Staves* before you go to a new strip having more than one staff and switch to *One Staff* immediately afterwards.

In the *Strips only View* the status is not relevant. For the fastest selection of control elements in the *Track Area* this view is recommended.

F10 + F11 Show ↔ hide mixer

F10 + X Toggle *MIDI View* ↔ *Strips only View*

F10 + F9 Set Mixer width → amount of strips

While holding down **F10** press **F9**. Release the keys.

Input **1, 2, 3, 4, 5, 6, 7, 8, 9, 0, Q, W, E, R, T, Y, U, I, O, P, A, S**

This will set the Mixer width for 1,2,3.....22 strips. Key layout:

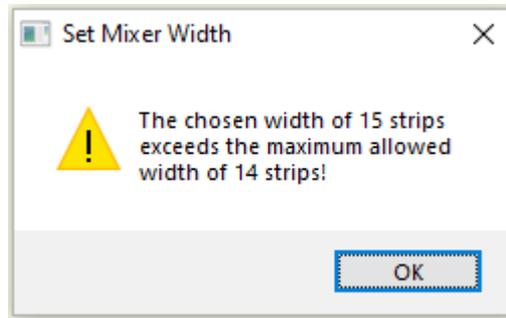
Q (11) under **1**, **W** (12) under **2** etc. **A** (21) under **Q** (11) and **S** under **W**.

When you have set the Mixer width on 1, 2 or 3 strips in *MIDI View* a tooltip appears to remind you that only strip 1, or 1 and 2, or 1, 2 and 3 of the four visible strips are active.

For instance after **F10 + F9** in *MIDI View* and input of **3**:



The width of the test screen makes a maximum of 22 strips possible. Wider screens could accommodate more strips. See below on page 15. If the width of your screen allows a maximum of say 14 strips and you would press **F10 + F9** followed by **T** this message appears:

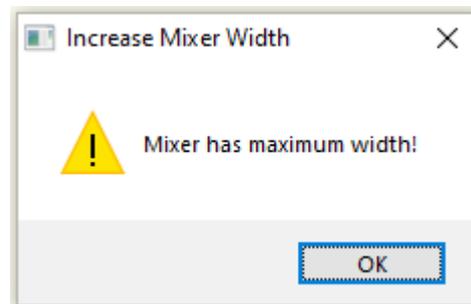
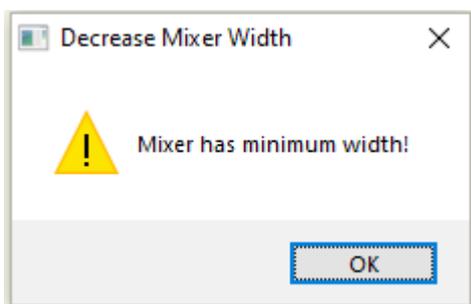


To let the computer know the maximum amount of strips you have to assign a value to the variable **MaxStripAmount** in the auto-execute section of the file. See below on page 15.

F10 + , (<) will reduce the window by one strip

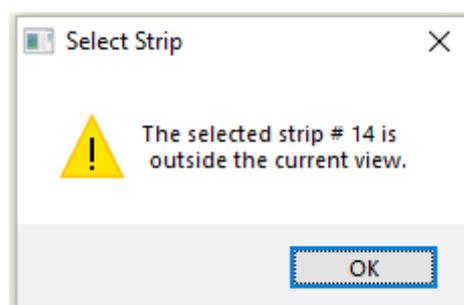
F10 + . (>) will enlarge the window by one strip

If you go too far one of these messages is visible for a short moment.



F10 + # Strip 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, Q, W, E, R, T, Y, U, I, O, P, A, S
Select element in this strip. See also the example given above in **Control + F10**. In *MIDI View* at status *One Staff* a click on *Pan* or *Slider* selects the strip. But when *Arrow, Solo* or *Mute* is selected the control element will be clicked without selection of the strip. Missing the target when the new instrument has more than one staff can then occur.

If you select an *allowed but invisible* strip a timed message pops up:



A not allowed strip however will by definition remain invisible.
If you select a not allowed strip you get this message:



Strip Navigation

There are more ways to move in and between strips than with **F10 + # Strip** or **F10 + 'Control Element'**. (See next section).

In the following hotkey combinations the second key was chosen because of its proximity to **F10**.

When you pass *Solo* or *Mute* the button changes state.

The next four hotkeys have alternatives which are a bit less user-friendly.

F10 + Insert	F10 + Up	Cycle through strip elements upwards
F10 + Delete	F10 + Down	Cycle through strip elements downwards
F10 + ;	F10 + Left	Select same element in left strip
F10 + '	F10 + Right	Select same element in right strip
F10 + [Scroll one strip left Left ↔ Right fast
F10 +]		Scroll one strip right Left ↔ Right fast
F10 + Home		Scroll to left edge - first strip
F10 + End		Scroll to right edge - last strip
F10 + Page Up		Scroll to previous strip page
F10 + Page Down		Scroll to next strip page

Control elements

F10 + F1 *MIDI View* Select sound

Select Instrument. Press Escape when ready.

F10 + F2 # *MIDI View* Reverb

Reverb. Enter Value.
Press Z when ready.

F10 + F3 # *MIDI View* Chorus

Chorus. Enter Value.
Press Z when ready.

F10 + F4 Show/hide part tracks

F10 + F5 Solo

F10 + F6 Mute

F10 + F7 # Pan

After terminating the input by pressing **Z** or **Escape** the mouse returns to the Slider.

F10 + F8 # Slider

F10 + F12 # Master Gain

Master Gain. Enter Value.
Press Z when ready.

F10 + / *MIDI View* Drumset

F10 + L # *MIDI View* Channel

Set Midi Channel. Enter Channel Number. Press Z when ready.

F10 + N *MIDI View* Partname

Enter the name. Press Escape when ready.

F10 + B # *MIDI View* Port

Set Midi Port. Enter Port Number. Press Z when ready.

F10 + V *MIDI View* Play Part Only

F10 + C *MIDI View* Track color

Choose Color. Press Z when ready.

F10 + M → *MIDI View* Mute Voice

Mute Voice. Enter Voice Number(s). Press Z when ready.

While holding **F10** press **M**. Release the keys and Input Voice Number(s).

Normal Staff **1, 2, 3, 4** or **F1, F2, F3, F4**

Piano Staff extra **5, 6, 7, 8** or **F5, F6, F7, F8**

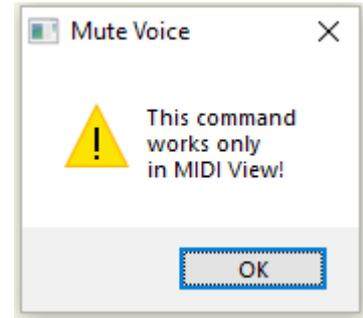
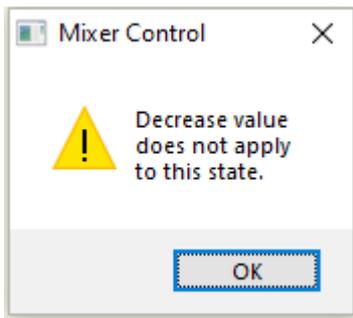
Organ Staff extra **9, 0, -, =** or **F9, F10, F11, F12**

For all numeric fields including *Master Gain*, *Pan* and *Slider*:

F10 + - or **Down** Decrease value

F10 + = or **Up** Increase value

Explaining messages:



Mixer variables in the auto-execute section

If you use Mixer.ahk as an independent group of macros the coordinates are part of the auto-execute section at the top of the file.

If the macros are part of the AutoHotkey Kit the macros reside in Master.ahk and the coordinates in Coordinates.ahk

In both cases there are some values of Mixer variables which are set in the auto-execute section to give them a starting value.

Changing the default number of strips

Change these lines in the auto-execute section:

```
ShowStrip := 10
```

```
StripSelect := 10
```

Change **10** into the number of your preference.

And in the coordinates section (or in Coordinates.ahk) change this line to match your new number of strips:

```
Mixer_Width := 924
```

Changing the maximum number of strips

Change this line in the auto-execute section:

```
MaxStripAmount := 22
```

If your screen is wide enough to accommodate more strips: there are (outcommented) hotkeys to *set* and to *select* max 25 strips.

After **F10 + F9** Input **D** - search for

```
If (SetStrip = "d")
```

```
    ShowStrip := 23 ; set width to 23 strips
```

And its counterpart

```
f10 & d:: ; select strip 23
```

```
StripSelect := 23
```

```
Goto, MixerStripSelect
```

General remarks

This script is published as a separate file: *Extra__Mixer.txt*. The extension must be renamed in *.ahk*. See page 6 of the pdf *AutoHotkey for MuseScore*. In order to read the non-ASCII characters correctly it has been saved as *UTF-8 with BOM*. See the FAQ in the AutoHotkey Help under 'Why are the non-ASCII characters in my script displaying or sending incorrectly?'

The **Z**-key is 'liberated' by redefining the MuseScore shortcut. See pages 9 and 40 of the pdf *AutoHotkey for MuseScore*.

The time MuseScore needs to open the Mixer window depends mainly on the number of staves. On the other hand the time needed to select a mixer control element does *not* depend on the number of strips.

The tool *PixelMousing* you find in the attachment to post 2 of the AHK Kit. *Window Spy* is part of the AutoHotkey installation.

When you want to use it as an independent group of macros: Remove the Play Panel items, change the path and enter your location of MuseScore in the auto-execute section at the top of the file.

When you want to *integrate the script in the AHK Kit*: The variables with the values valid for the test screen are in the auto-execute section at the top of the file. Move them to *Coordinates.ahk* and enter there the values found by you. Move the part *Mixer Initialisation* in the auto-execute section to the auto-execute section of *Master.ahk*. Copy the Mixer and Play Panel macros to your *Master.ahk*. The Mixer macros also include the utility **Z + 6** needed to find the values of **Instr_Status**.

Changes in the Master file of the AHK Kit: Some Info screens of the *Master.ahk* have been extended to include references to the Mixer macros. These are:

- [+ **Z** Zoom and View. See *Extra__Selection_and_Navigation.txt*
- [+ **M** Master More in *Extra__Mixer.txt*. See page 19 *Play Panel Hotkeys*.
- [+ **U** Utilities. Extended to include the command **IMXR** *Initialise Mixer*
See the next chapter.

Adjusting the position of the Tooltips

Dependent on your screen resolution you probably have to change some coordinates. In the macros these lines are marked with (***change***).

Initialise Mixer

Z + U Master file *Utilities* → Inputbox **IMXR**

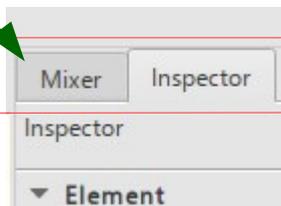
After launching MuseScore in some setups the Mixer window is displayed docked within the Inspector. Also if the docked window is in *MIDI View* the Inspector is not in the Defined State because it will be too wide.

The Mixer must be detached from the Inspector before it is fit for use by the Mixer macros and the Inspector must be restored to its defined width.

The procedure is comparable to that of the initialisation of the Play Panel.

We need two images: **IM_30_Mixer.png** and **IM_31_Mixer_Inactive_Tab.png**

 and  The difference is the slightly darker coloring of the second image where *Mixer* is the title of an inactive Tab usually next to the active Tab *Inspector*.



The X-coordinates of the search surface are from half screen width to the right edge of the screen which allows for a very wide mixer.

The Y1 describes the upper red line.

The Y2 describes the lower red line.

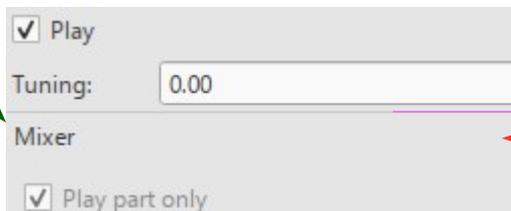
IM_31_Y1 := 22

IM_31_Y2 := 46

ImageSearch, X, Y, A_ScreenWidth / 2, IM_31_Y1, A_ScreenWidth, IM_31_Y2, IM_31_Mixer_Inactive_Tab.png

When the image is found the mouse clicks the upper-left corner which activates the Tab. The title changes into its slightly lighter variant which will now be searched.

When the *Inactive Tab* image is not found the search for the lighter variant starts immediately. If this variant is found there follows a double click to undock it from the Inspector. Now the Mixer can be moved to the position described in the Mixer macros. Just as with the Play Panel we'll need an offset to avoid the **magenta** horizontal and vertical separators.



Mixer_Off_X := 250

Mixer_Off_Y := 10

The double click is on the point of the red arrow.

When the Mixer is not present at all - the lighter variant is not found - it will be activated via the *View Menu* followed by the Image Search for , the double click and the positioning of the Mixer window as described above.

Finally the width of the Inspector will be reset to its defined width. The macro follows the same procedure as in the hotkey **Z + 2**. See page 23 of the pdf *AutoHotkey for MuseScore* and **IM_01_2_Inspector_Displaced_Insp.png**.

AutoHotkey Kit for MuseScore - Play Panel revisited

First description of the **Initialisation**: *AutoHotkey_for_MuseScore.pdf* page 48
Notice there `Play_Panel_X`, `Play_Panel_Y`, `PlayP_Wide` and `PlayP_High`.

Z + U ► IPP Utilities → Initialise Play Panel

The first version did not reckon with the possibility that the Play Panel is docked in the Inspector as an Inactive Tab. The updated version follows a method a bit comparable with the initialisation of the Mixer described above.

 Already in original version: `IM_19_Inspector_Play_Panel_Header.png`

 Added `IM_32_Play_Panel_Inactive_Tab.png` slightly darker coloring



Positions of the Play Panel relevant macro lines are marked (***change*?**)

Panel is *above the Inspector*. The upper-left corner (point of green arrow) is a tiny bit higher than when found as active tab. The macro has a MsgBox which shows the height in this situation. Determine your height in `If (PP_Y = 24)`.

Panel is *active tab*. The macro has thus determined that the Panel is *not* above the Inspector. The black arrows describe the height in which the Header can be found as a tab. The Y-coordinate of the top arrow point is 0, the upper edge of the screen. In the shown case the Defined State is Max Canvas, full screen, no toolbars. in `If (PP_Y < 40)` determine your number in your Defined State instead of `40`. Because the header has been found as active tab the clickpoint to undock the Play Panel must be lower than in the more common situation where the Play Panel is above the Inspector or somewhere below the Inspector. In `Coordinates.ahk` we have `PP_Off_Y := 4` for this more common situation. Determine your number for the active tab case. `PP_Off_Y := 29` and enter it in the macro in the lines marked by (***change*?**)

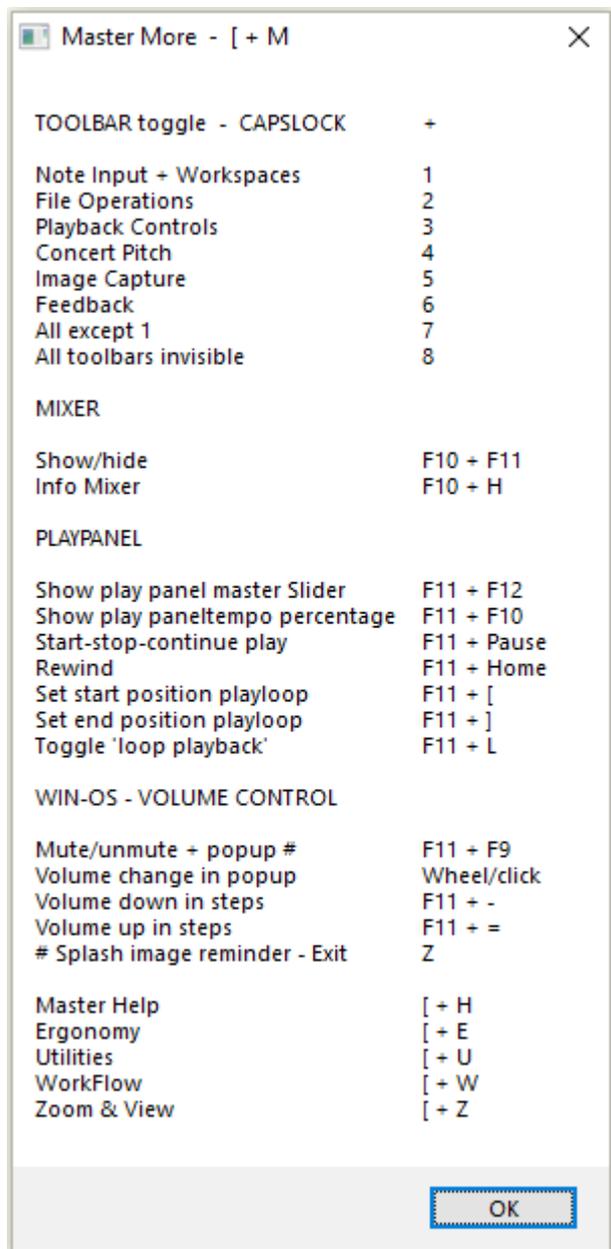
Panel is *inactive tab*. By a click on the found image it becomes active and ready to be treated as an active tab.

Panel is *below the Inspector*. It is treated in the same way as *above*.

Panel is *not present*. It will be shown via the *View Menu*. Now it can be an *active tab*, *above* or *below* the Inspector and will be treated as such.

Play Panel - hotkeys

[+ M The Info screen Master → More shows the commands



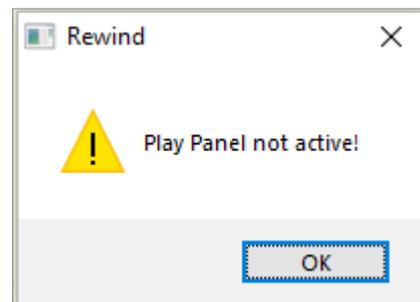
New hotkeys:

F11 + Pause

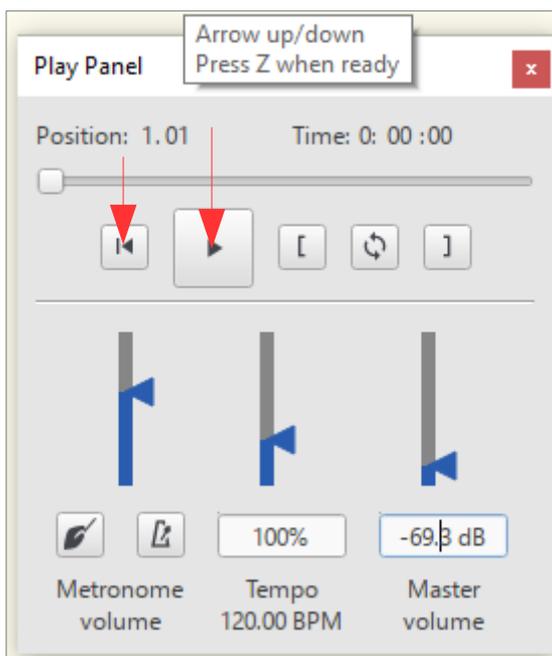
PP_Play_X := 104
PP_Play_Y := 106

F11 + Home

Rewind_X := 60
Rewind_Y := 106



The loop hotkeys use custom MuseScore shortcuts. See below.



F11 + [~f11 & [::Send !+i Alt + Shift + I
F11 +] ~f11 &]::Send !+o Alt + Shift + O
F11 + L ~f11 & l::Send !+l Alt + Shift + L

Replace the **Send** commands by your custom shortcuts.

The ergonomical advantage is in the proximity of the used keys.