

User Guide

Ver 4.0

Table of Contents

Introduction.....	3
What is a scene?.....	3
Commands.....	3
Save	3
Restore	3
Fade	4
X-Fade.....	4
User Interface.....	5
Interactive mode	5
Launching snippets (option -a).....	7
Buttons & snippets from file (option -F)	7
Script mode	8
SAFES	9
MIDI control	11
Examples.....	11

Introduction

wxfade is a remote tool running on Windows based computers for managing WING scenes.

WING scenes are handled as binary files and are kept in the same directory the program runs from. The program is launched in a command-line or terminal window and will automatically find and connect through network to the WING console in the same /24 local network¹ as the computer.

wxfade enables several operations with scenes, presented in the chapters below.

What is a scene?

In the case of **wxfade**, a scene is a set of binary nodes collected from the WING console, and saved as a single binary file. A scene is composed of all the attributes and settings for the following elements: /ch/1-40, /aux/1-8, /bus/1-16, /main/1-4, /mtx/1-8, /dca/1-16, and /fx/1-16. In the current version, **wxfade** scenes do not include any routing, extension or card settings as these are unlikely to be modified during a single show. It also optimizes the size of files and enables a faster update from a remote application. With a freshly initialized WING, each scene file is approximately 125kB.

Commands

wxfade provides the following functions which can be used in interactive mode (i.e., directly from the console) or as single-shot script commands. All functions and parameters can be controlled through MIDI controllers if needed.

Save

The **Save** command saves the current state of WING² to a binary file on the remote computer, identified with an associated scene number.

Restore

The Restore command loads the binary file associated with a scene number and updates the WING state with the data contained in the file. The restore action is immediate and optional **SAFES**³ can be set to prevent modifications of some of the WING channels, as shown below under **SAFES**.

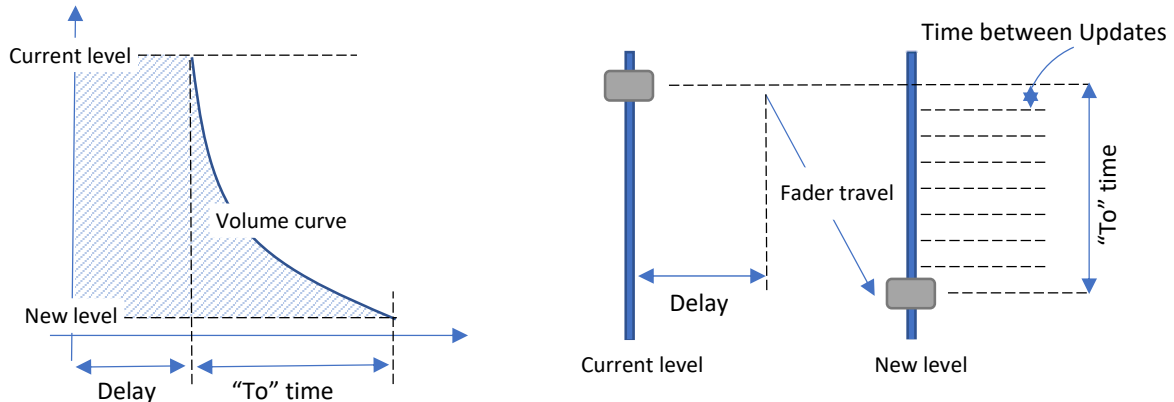
¹ For ex: if your network is 192.168.0.1, the IP range 192.168.0.xxx will be scanned for a WING reply

² /ch/1-40, /aux/1-8, /bus/1-16, /main/1-4, /mtx/1-8, /dca/1-16, and /fx/1-16

³ See later in this document

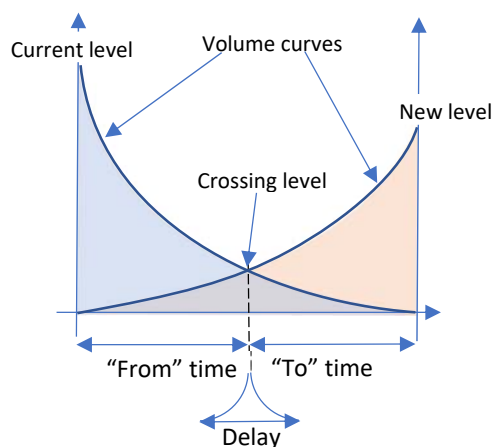
Fade

The **Fade** command is acting as a **Restore** action, with a smooth change in levels from the current state of the console to the final level values contained in the restored data. An optional Delay can be applied after scene update and before level changes take place. Optional **SAFES** can be active when restoring a scene, filtering out some of the data. The Delay, "To" time and Update rate of level changes can be modified with optional parameters; The New level could well be higher than the Current level, changing the shape of the curves shown below:



X-Fade

The **X-Fade** command is also acting as a **Restore** action, with smooth level changes from the current state to the final level values contained in the restored data, and respecting travel curves corresponding to different travel times applied to the Current level and New level to and from a Crossing level value, as shown below. The actual scene update occurs at the time when levels get to the Crossing level value. An optional delay can be applied at that time. **SAFES** can be active when updating the scene, filtering out some of the data. The Crossing level, Delay, "From" and "To" times and Update rate of level changes can be modified with optional parameters, as shown below. Note the Crossing level can be set to any value between -144dB and 10dB; The curves shown below as example could be very different:



User Interface

The following paragraphs show how to interact with **wxfade** on the console, or from the command-line terminal using scripting options.

Interactive mode

When launching **wxfade** with the **-i** option, the interactions will take place directly at the console, using CUSTOM CONTROL knobs and buttons from a chosen layer. By default, **wxfade** uses layer 2, and the first 3 columns of knobs and buttons. These can be changed using options **-l** [layer: 1..16] and **-b** [buttons 1..2].

As **wxfade** is running, the following text/options are shown after you select CUSTOM CONTROL Layer that receives the application GUI [default value: 2]:



Scene **Save** or **Restore** take place after you turn the leftmost knob to display a scene number, and press the respective button below that knob. The **Save** or **Restore** action are immediate; no confirmation is asked for.



Fade or **X-Fade** can be chosen by selecting the appropriate item in the 3rd column.

The choice of values for the respective parameter is done by using the 3rd column knob. The pictures below show the list of parameters that can be selected using the 3rd column buttons, *scroll_up* or *scroll_dn*:

Turning the knob will toggle between “**Fade**” & “**X-Fade**”

Fade	X-fade
scroll_up scroll_dn	scroll_up scroll_dn

Additional parameters include:

F_t: 2000	T_t: 1000	C_v: -144	C_t: 0	U_t: 30
scroll_up scroll_dn	scroll_up scroll_dn	scroll_up scroll_dn	scroll_up scroll_dn	scroll_up scroll_dn

F_t: the time spent moving from the Current level to the Crossing level. The value is shown in ms and can be set between 0 and 12700. This is only used with **X-Fade**.

T_t: the time spent moving from the Current level to the New level when using **Fade**, or from the Crossing level to the New level when using **X-Fade**. The value is shown in ms and can be set between 0 and 12700.

C_v: The Crossing Level value. The value is shown in dB and can be set between -144 and 10. This is only used with **X-Fade**.

C_t: The time set as a wait delay right after loading the new scene; Using **Fade**, this is right after launching the change request, or arriving at the Crossing level when using **X-Fade**; The value is shown in ms and can be set between 0 and 12700.

U_t: the time in-between fader updates, setting how smooth fader moves will be. The value is shown in ms and can be set between 0 and 127.

Fade or **X-Fade**, going from the current WING state to a new scene with smooth fader moves takes place by selecting the new scene using the 2nd knob, or by using the Next/+1 button under that knob and finally pressing the GO! button; The parameters are applying as presented above.



Launching snippets (option -a)

When running in interactive mode, **wxfade** offers 3 options: `-a`, `-o`, `-z` dedicated at launching snippets.

The `-a` option⁴ is used to set and name a specific button of the user choice (button number and layer).

Options `-o` and `-z` are respectively used to associate a node (or snippet) of max 128 characters that will be sent when the button toggles from 0 to a value other than 0. By default, the button selected with option `-a` is set to 0 and if it exists, the node corresponding to that value is sent to the console when the program initializes.

If set with option `-z`, the associated node snippet will be sent to the console at startup, whatever mode the programs will run (interactive or not).

Errors can be reported while using options: `-a`, `-o`, `-z`; Should this happen, the program will terminate.

Buttons & snippets from file (option -F)

The options: `-a`, `-o`, `-z` proposed above are great when only a few buttons/snippets are launched from the command line. It is much easier to use an organized file for setting up buttons and associated ON and OFF snippets when larger numbers of buttons are involved. **wxfade** ver 3.0 and above can manage up to 128 buttons per file, each button associated with an ON and an OFF snippet. This offers a lot of flexibility and puts the Custom Control Central section to good use.

Option `-F` is used to pass on a file name⁵ to the program. When scanning for that option, **wxfade** will parse the contents of the file and cumulates the definitions found in the file with the already data passed on with `-a`, `-o`, `-z` options if any.

Note that new buttons can replace already defined ones. The user is responsible for ensuring there is a logic in the button assignments.

The file containing buttons and snippets definition is a simple text file (with a `.txt` extension) with each line containing a single button definition. The elements are separated by spaces. As a result, space characters are not allowed in name or snippets. The example files provided with the `wxfade.zip` package contain further details on how to populate the file, describing the possible values and maximum lengths for string-like components. In the file, lines beginning with a '#' are ignored and treated as comment lines. Errors can be reported while parsing files for button data; Should this happen, the program will terminate.

As in the case for option `-z`, node snippets from the file associated with a button's OFF case will be sent to the console at startup, whatever mode the programs will run (interactive or not).

The selected file can reference another file, setting the number of controllable buttons and snippets to infinity. A new definition file can be loaded on the press of a button; The user can create all sorts of menus using this neat feature, or enable modifications to the buttons while the program is running, by enabling the current file reload.

⁴ See actual description/usage in the "Script Mode" chapter

⁵ See actual description/usage in the "Script Mode" chapter

Script mode

Script mode takes place when running **wxfade** without the **-i** option. It can be quite useful when including **wxfade** commands in another program, such as for automation for example. The **-q** option can be used to tell **wxfade** to be totally 'silent' (i.e., not printing any text, except for errors). The **-h** option can be used to print a quick reminder of all possible options, resulting in the text below being displayed:

```
usage: wxfade [-i, to launch interactive mode]
              [-q, keep quiet (no messages other than errors)]
              [-l 1..16 (default: 2), set choice of layer]
              [-b 1..2 (default: 1), set choice of buttons & knobs set]
              [-s 1..127, save state to scene file 1..127]
              [-r 1..127, restore state from scene file 1..127]
              [-x 1..127, cross fade from current state to scene file 1..127]
              [-n 1..127, simple fade from current state to scene file 1..127]
              [-f 0..12700 (default: 2000), time in ms to 'Crossing' value]
              [-c -144..10 (default: -144), 'Crossing' value/level]
              [-t 0..12700 (default: 1000), time in ms from 'Crossing' value]
              [-w 0..12700 (default: 0), wait time in 'Crossing' value during cross fade]
              [-u 0..127 (default: 30), level update rate in ms]
              [-a <L,B,u|d,name>, associates button B layer L to be button up or down called 'name' (16 c max)]
              [-o <snip1>, node data snip1 (128c max) to be sent when the button set with -a is != 0]
              [-z <snip0>, node data snip0 (128c max) to be sent at init and when the button set with -a is 0]
              [-F <file>, populate -a/-o/-z options from file - see documentation for format]
              [-S <safes>, safes is a list of channels safe from restore:
                  ex: /ch/1,/ch/5-10,/mtx/1-4
                      ch 1, 5 to 10 and matrix 1 to 4 will not be restored]
              [-m <ch,cc1,cc2,cc3,cc4,cc5,cc6,cc7,cc8,cc9>, changes default MIDI settings, with ch = MIDI channel]
                  cc1: 119 - MIDI cc for 'scene' number [1..127]
                  cc2: 120 - MIDI cc for 'Save' on value 127->0
                  cc3: 121 - MIDI cc for 'Restore' on value 127->0
                  cc4: 122 - MIDI cc for 'Go_To' number [1..127]
                  cc5: 123 - MIDI cc for 'GO!' on value 127->0
                  cc6: 124 - MIDI cc selecting 'Fade, X-fade, F_t, T_t, C_v, U_t, C_t' values [0..127]
                  cc7: 125 - MIDI cc scrolling up in cc6 list on value 127->0
                  cc8: 126 - MIDI cc scrolling down in cc6 list on value 127->0
                  cc9: 127 - MIDI cc for scene + 1 on value 127->0
                  cc10: 0+j - MIDI cc for button #j set with option -a on value 127->0
```

In interactive mode (option -i), wxfade subscribes to WING events on Custom Control Central Encoders & Buttons chosen with options -l and -b.
You can set a scene number using the first knob; the two buttons underneath are to save and restore a scene.
You select a scene for Simple- or Cross- fading using the 2nd knob or the 'Next/+1' button underneath
The button 'GO!' launches the fade action.
You set Cross- or Simple- fade parameters with the 3rd knob; Buttons underneath scroll up/down in the list of parameters
Option -i is needed along with options -o, -b, -l, -F or -m
Option -a is needed along with and must precede options -o or -z
Option -s <num> launches an immediate save to scene <num>
Option -r <num> launches an immediate restore from scene <num>, using -S option values
Option -x <num> launches a one-shot Desk cross fade to scene <num>, using options -c, -f, -t, -u, -w, -S values
Option -n <num> launches a one-shot Desk simple fade to scene <num>, using options -t, -u, -w, -S values
Options -i, -r, -s, -x, -n are exclusive of each other
The Files WING_bScene_0..127 are binary files located in the same directory as the program

The options are offered to execute the **Save**, **Restore**, **Fade** and **X-Fade** commands described above, and using their respective parameters:

Save: The **-s** option will save the current WING state to a scene which number is provided with the option. Scene numbers can be [1...127]. No other parameters are required.

Restore: The **-r** option will set the WING state to the values contained in a scene which number is provided with the option. Scene numbers can be [1...127]. The **-S** option can be used to set a list of

SAFES preventing some of the channel strips or strip elements from being affected by the Restore command. No other parameters are required.

Fade: The `-n` option will set the WING state to the values contained in a scene which number is provided with the option, with a smooth update of level values from the current scene to the new one. Scene numbers can be [1...127]. Parameters provided with the `-w`, `-t` and `-u` options respectively set the delay to wait before the levels start to change, the time sweeping from the Current to the New levels, and the Update rate at which levels are changed. The `-S` option can be used to set a list of **SAFES** preventing some of the channel strips or strip elements from being affected by the Fade command. No other parameters are required.

X-Fade: The `-x` option will set the WING state to the values contained in a scene which number is provided with the option, with a smooth update of Current level values to a Crossing level value. The new scene data is loaded and the program proceeds with an optional delay followed by smooth level transitions from the Crossing level value to the New level values. Scene numbers can be [1...127]. Parameters provided with the `-f`, `-t` and `-u` options respectively set the time from the Current level values to the Crossing level, the time from the Crossing level to the New level values, and the Update rate at which levels are changed. The `-w` and `-c` options represent the delay to wait for right after loading the new scene, and the value of the Crossing level. The `-S` option can be used to set a list of **SAFES** preventing some of the channel strips or strip elements from being affected by the X-Fade command. No other parameters are required.

SAFES

With the **Restore**, **Fade** and **X-Fade** operations in interactive or scripting modes, **SAFES** can be applied in order to protect some channel strips or strip elements from being modified with new scene data. The list of channel strips or strip elements to protect from scene updates is provided when launching **wxfade**, using the `-S` option. The **SAFES** list is made of strings separated by a comma, each representing a channel strip or a contiguous list of channel strips and possible sub-elements to protect.

Besides '*all-parameters*' **SAFES** (by listing the strip only), **wxfade** can 'safe' **fdr**, **mute**, and **solo**⁶ by adding the respective element to the strip as shown below for fader level and mute.

For example, the following list:

```
/bus/1-3/mute,/bus/2/fdr,/ch/2-3,/ch/4-5/fdr,/ch/5-6/mute,/ch/1/solo
```

will protect complete channels 2 and 3; Additionally, mutes for bus 1 to 3 and channels 5 and 6, and fader moves for bus 2 and channels 4 and 5, as well as channel 1 solo will be protected too. All other channels/elements will be updated with the new scene being loaded to the WING.

⁶ Text beginning with '\$' is considered a shell variable under several operating systems and therefore will not be passed to the application as a text string, the value "solo" will be preferred over "\$solo".

The -S option can be provided when launching **wxfade**, and can only be modified as the program is running if placed in a file loaded during program execution (by setting a file name to be loaded with the press of a User Control button). In that case the new **SAFES** are gathered from a single line starting with '-S'. Each occurrence of a '-S' line completely replace all existing **SAFES** with new ones. A line containing only the string "-S" will remove all **SAFES**.

The following lines show a possible setup:

Launching the program in interactive mode with `file1` being loaded, no safes apply.

```
<your prompt>/wxfade -i -F file1.txt
```

`file2.txt` contains the following lines as part of its contents:

```
[...]  
# Line below: button 4-1-up to load a whole new set of SAFES.  
4 1 u SetSafes file2.txt  
[...]
```

Now let's have `file2.txt` contain the following lines as part of its data:

```
[...]  
# Set new safes  
-S /ch/1,/ch/6-7/mute,/ch/9-11/fdr  
#  
[...]
```

Button UP, layer 4 column 1 will look like this:



And pressing the SetSafes button will load `file2.txt`, resulting in new SAFES into action:

Ch 1 will be fully safe, while Ch 9 to 11 faders and Ch 6 and 7 mutes will be safe.

MIDI control

There are times when it can be necessary to control the application through MIDI, or using a MIDI tool that can communicate with the WING for other purposes. Such operation takes place while running **wxfade** in interactive mode. Every action or parameter described above can be set or modified using 10 distinct MIDI continuous controllers (CC) and a MIDI channel. The default **wxfade** values of MIDI channel 16 and MIDI CC 100 to 109 can be changed using the **-m** option, followed by all ch and cc1...cc10 values separated by a comma. Depending on the parameter, the respective controller value will take any value between 0 and 127, or be active on a value transition from 1..127 to 0. Details are provided when running **wxfade** with the **-h** option.

Examples

The following screenshots are showing different runs of the **wxfade** application⁷.

Simple launch in interactive mode; Actual modifications and actions will take place from knobs and buttons being activated at the desk.

```
PS C:\...\release> ./wxfade -i
--wxfade-- ver 2.0 @2022 Patrick-Gilles Maillot
Using wapi ver: 1.24
WING found at IP: 192.168.0.11
```

Simple launch in interactive mode, with SAFES protecting all /main channels; Actual modifications and actions will take place from knobs and buttons being activated at the desk.

```
PS C:\...\release> ./wxfade -i -S /main/1-4
--wxfade-- ver 2.0 @2022 Patrick-Gilles Maillot
Using wapi ver: 1.24
WING found at IP: 192.168.0.11
```

Simple launch in script mode, saving scene 10. Action is immediate.

```
PS C:\...\release> ./wxfade -s 10
--wxfade-- ver 2.0 @2022 Patrick-Gilles Maillot
Using wapi ver: 1.24
WING found at IP: 192.168.0.11
Saved scene 10
PS C:\...\release>
```

⁷ Video demos are available at <https://x32ram.com/videos>

Same as above, quiet mode requested, saving scene 10. Action is immediate.

```
PS C:\...\release> ./wxfade -s 10 -q
PS C:\...\release>
```

Launch in interactive mode, with `-a`, `-o`, and `-z` options used to set button 4 layer 2 to “ToggleTest” and use two separate snippets to be sent when button is toggling from 0 to ~0.

```
./wxfade -i -a 2,4,ToggleTest -o /aux.8.fdr=0,mute=0 -z /aux.8.fdr=-40,mute=1
--wxfade-- ver 2.1 @2022 Patrick-Gilles Maillot
Using wapi ver: 1.25
WING found at IP: 192.168.0.11
```

A demo video of the above `-a`, `-o`, `-z` options can be seen at

<https://sites.google.com/site/patrickmaillot/wing#h.30vtyc1kws44>

Launch in interactive mode, with `-a`, `-o`, `-z` and `-F` options, adding the buttons and nodes contained in a text file.

```
./wxfade -i -a 2,4,ToggleTest -o /aux.8.mute=0 -z /aux.8.mute=1 -F testfile.txt
--wxfade-- ver 3.0 @2022 Patrick-Gilles Maillot
Using wapi ver: 1.25
WING found at IP: 192.168.0.11
```