



Super Media Slide Show User Guide

V1.7.2

for OBS-Studio

by
Keith Schneider – Spyratech.com

A note about this document.

In case you ask the question, Where is the source for this PDF? Answer – Its right here, embedded in this PDF file. This document was prepared using the LibreOffice suite of software on a Linux desktop system. The original document started out as a file with the standard .odt (open document type) file type. When the finished document was exported to the PDF formatted file for distribution, the source .odt document was embedded into that PDF file. Therefore, to change this document, use LibreOffice to open the PDF file and the original .odt source document will be automatically and instantly extracted and brought up in the LibreOffice word processor ready for edit. You can save the document back to a source .odt file if and whenever you wish. When you complete your modifications to the document, you should then export it again to PDF format and ensure that the source .odt document is once again embedded into the PDF file (don't forget! - there is a [embedded ODF file] check box in the export to PDF dialog) so the final PDF and original .odt document are perfectly kept together. LibreOffice is a complete, comparable and highly competitive package to Microsoft's Office suite and is totally Free & free. The LibreOffice Suite is available for Linux, Windows and MacOS at: <https://www.libreoffice.org>

Table of Contents

1 INTRODUCTION.....	3
2 LICENSE.....	4
3 INSTALLATION.....	4
4 DEFINITIONS.....	5
A “SHOW” DEPENDS ON PERSPECTIVE.....	5
RELATIVE PATHS AND WINDOWS.....	5
MEDIA VS VIDEOS.....	5
5 HOTKEYS – SAFE-TERMINATE.....	6
6 ORGANIZING SOURCES.....	7
7 PROPERTIES.....	8
SCENE AUTO START.....	9
SHOWS DISABLED.....	9
PICTURE VIEW TIME.....	9
GROUP SOURCE.....	9
TEXT SOURCE.....	10
IMAGE SOURCE.....	10
MEDIA SOURCE.....	10
BACKGROUND AUDIO SOURCE.....	11
BACKGROUND AUDIO CUT.....	11
BACKGROUND AUDIO FADE TIME.....	11
BASE FOLDER PATH.....	11
SCENES LIST.....	12
SCENES LIST FLAGS.....	13
RECORDING CONTROL DIRECTIVES – THE ! CHARACTER.....	14
FOLDER PATH TRIM LEVEL.....	15
SLIDE SHOW LOOP.....	15
RANDOMIZE SHOW.....	15
THE SHOW SAFE TERMINATE BUTTON.....	15
DEBUG ENABLED.....	15
8 MULTI-SCENE EXAMPLE.....	16
9 SUPPORT SERVICES.....	17
10 MISCELLANEOUS.....	17
MEDIA COLLECTION COMMANDS – WHY?.....	17
CURRENT WORKING DIRECTORY – THOUGHTS.....	17
SOURCE TRANSITIONS – THOUGHTS.....	18
FUTURES.....	18

1 Introduction

This document is a guide to the setup and usage of the Super Media Slide Show lua script for OBS-Studio.

SMSS is a lua script that drives the OBS-Studio Image and or Media source to deliver a well featured Multi-Media Slide Show like presentation. Media files are loaded from folders on your system and there is little limit to their number (thousands ought be no problem).

Features:

- Shows Images - The simple most obvious core of a slide-show. Any of the image types supported by the OBS-Studio Image source can be used and an Image view duration time is user selected.
- Plays Videos - Just like showing pictures, Videos play but they play for the duration of the Video while Pictures show for a designated time period. Any of the media types supported by the OBS-Studio Media source can be used although Video/Visual types are the most logical.
- Managed Background Audio. A Background Audio source volume can be faded by percent while Videos are shown. Volume is transitioned back to normal when Video(s) complete.
- Near unlimited Media files possible. Media files are loaded between source deactivation's / activation's. Original design intent was to support thousands.
- Multi-Scene work-flow like capabilities. Each Media File Collection entry is associated with a Scene and each one can optionally define a Next-Scene. This enables the creation of an automated switching from scene to scene for as many entries as defined.
- Co-exists with OBS-Studio configured Show/Hide transitions. Image visibility duration takes into consideration Image Source show/hide transition times.
- Optionally shows each full/partial file specification in a Text source during a show.
- Capable of Starting and Stopping Recording Automatically. This feature also acts to Fade the optional Background Audio source in at recording start and fade it out at recording end.
- **New in V1.7.0,** Scene Flags settings allow flexible Image view times, Audio Fade times and percentages, Hide Media File Paths (text source), Randomize slides etc.

All this and more. This script was developed for Linux and tested on Windows. Aside from platform specifics, it works identically on either. MacOS is expected to just work.

The basic inspiration for this software is multifaceted. First, came the discovery of OBS-Studio and then came the an understanding of limitations in the existing Image-Slide-Show source in OBS-Studio and other efforts to overcome the same. I have also been programming since the DEC PDP-11 (16 bit, 64KB address space, a few hundred thousand Instructions per second, punched paper tape boot loaders) days (about 45 years) and am now largely retired. I still itched to program but wanted something worthy and interesting to latch onto. I particularly wanted to take thousands of pictures, scanned from many boxes of old family pictures and make them available to the family in a format that they would actually enjoy. This would allow the collection to live on vs these old boxes of things being stored and forgotten.

Thus, the SuperMediaSlideShow script program for OBS-Studio came into being. The overall experience has been an enjoyable indulgence in Lua as well as doing something different.

Lastly, I want to extend a gigantic multitude of Thanks to all in the OBS-Studio lua community from which I was able to draw on tips and tricks to get things done.

2 License

This software is licensed under the terms of the:

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

COPYRIGHT NOTICE:

Copyright (c) 2022 Keith A. Schneider

Copyright (c) 2022 SpyraTech International LLC

An unpublished work.

The text of this license accompanies this software and is implicitly incorporated into the body of the software.

3 Installation

The Super Media Slide Show software is distributed as a zip file. Download the file and unzip it into a suitable location on your system that you are least likely to change. You should have your unzip application preserve folder paths when unzipping. The result should be a folder and a set of files that appears as follows:

```
SuperMediaSlideShow/README.md
    SuperMediaSlideShow.lua
    SuperMediaSlideShow.pdf
    locale/en-US.ini
    license/gpl-3.0.txt
```

To install the script into OBS-Studio, start OBS-Studio and:

- Select Tools-->Scripts from the top menu bar.
- In the Scripts main window, lower left, select the (plus) icon.
- Use the provided file dialog box to navigate to the SuperMediaSlideShow folder you installed from the zip package.
- Locate and select the SuperMediaSlideShow.lua script file.

The script will appear in the list of available scripts known to OBS-Studio. On the right side of the Window, you can observe the SuperMediaSlideShow properties layout. You can then proceed to fill out your slide show properties and setup your sources etc. Usually, it is best to get the needed Sources setup before you configure the settings.

The default installation disables the starting of any shows. When you have your sources and overall configuration(s) sufficiently setup to start testing and running shows, you will use the SMSS properties settings to Enable Shows. Note also that when a show completes and you want to do further changes and or navigation through Scenes and Sources, it is best to go to the settings page and Disable Shows. This prevents scenarios where moving through scenes may inadvertently cause a Slide Show to try to start. Form this habit early. Perhaps a hotkey to do this will come later but without a visual means of showing the user that Shows are Disabled, I have been reluctant.

4 Definitions

There are few definitions, outside of properties/settings, that need specific documentation herein, but here you will find items that are considered important.

A “Show” depends on perspective

In the heart of SMSS, a “Show” is largely meant to be understood as a single collection of Images & Media. It may be best thought of as a single entry in the settings “Scenes List”. Multiple entries are each treated as a collection of single “Shows” that happen to be chained together to make what may appear in a grand perspective, a “Show”.

A SMSS “Show” has to start up, present the Media collected and then shutdown. Each time, it has to find the sources and set them up et al and use them to show the Media and then unwind the initial setup process.

The “Scenes List” entries may be written to chain Shows from one to another but, again, the heart of SMSS treats them as individual shows and tries to use these Entry parameters such as the Next-Scene and certain uses of the “!” characters as tools to string shows together as seamlessly as it can.

Relative Paths and Windows

The Windows DIR command allows a relative path specification in the command but always returns absolute file paths. Nonetheless, the feature of providing the Base Folder Path in SMSS settings, remains a plus.

Media vs Videos

Because in OBS-Studio, the built-in source that plays Media is called the “Media Source”, I’ve tended to latch onto that terminology and you will find it used herein. But for all practical purposes, while the term Media is generalized to potentially mean that it can also play Audio files (ie: mp3), for the real goals of the SMSS software, the expectation is that you are configuring the “Media Source” for purposes of playing prerecorded Video content.

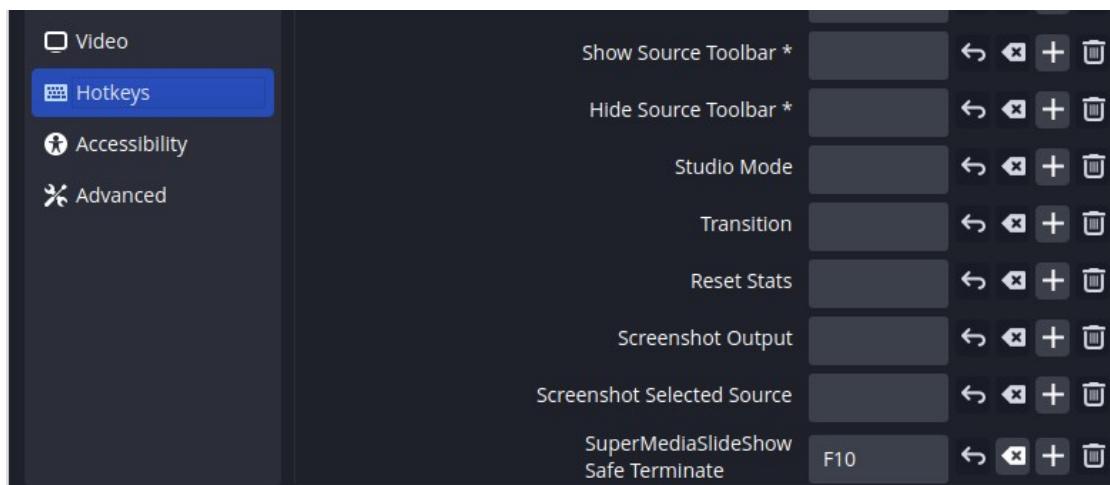
5 HotKeys – Safe-Terminate

When a slide show runs, especially while you are getting your scenes and sources all setup you will be running test shows and often only partial shows. In this scenario, a partial show is one that starts and runs only as long as you let it run and you stop it by preferably using a defined HotKey.

When the HotKey is pressed, this just signals to the program that it should immediately act like it has reached the end of the list of Media files for this collection and terminate (even if the show is set to Loop Continuous). When a show ends this way, the SMSS program will immediately terminate the current Slide and then stop the show in an orderly fashion and restore each source back to its original state, ready for a next show activation.

You will probably find out that the HotKey is a nice way to stop a show. That said, you can stop a show by simply clicking on the Show Group Icon, which in turn stops each source within the group. This is an abrupt stop and can leave each source in a somewhat indeterminate state (Visible y/n, Active y/n etc). It's generally harmless as the SMSS lua script tries to restore original source states when it stops. Also, the SMSS lua script can on-the-fly auto fix such confused sources when it acts to start another show. It was this original confusion of the sources that came about the name of Safe-Terminate for this HotKey. The name lives on but the importance of using it vs killing a show abruptly is now less.

You can define the HotKey to any key you wish. I tested with the F10 key. The example setup shown below is for that Key. **Navigate to the Main Toolbar and select File-->Settings and then select Hotkeys.** Search through the list for the entry labeled as SuperMediaSlideShow Safe Terminate.

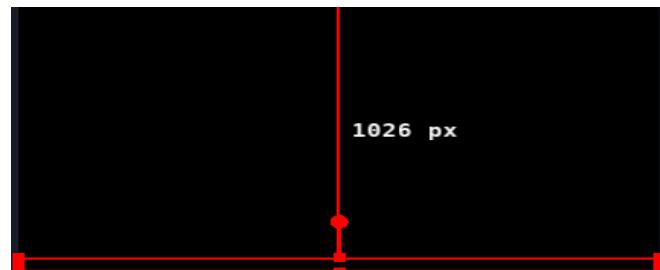


6 Organizing Sources

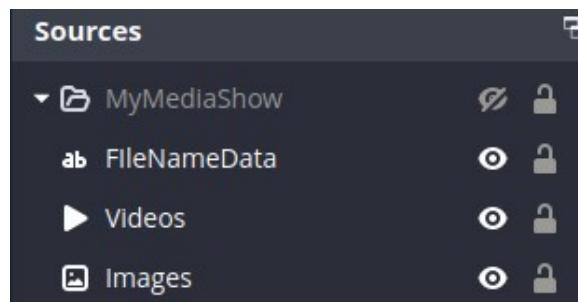
A number of sources are needed to form a slide show. How you organize them within the screen for your show presentation is up to you. When the Group setup shown herein is complete, you can then, when ready, add this group to additional scenes if you want to make a multi-scene show. A simple and generic configuration is as follows:

1. Create a Scene

2. **Add a Text source.** Setup the size to be the full width of the screen and only tall enough to hold one line of Text. Position this source at the bottom of the screen. You can initially enter some test text into the source and select a font, font size, color etc. eg: FileNameData



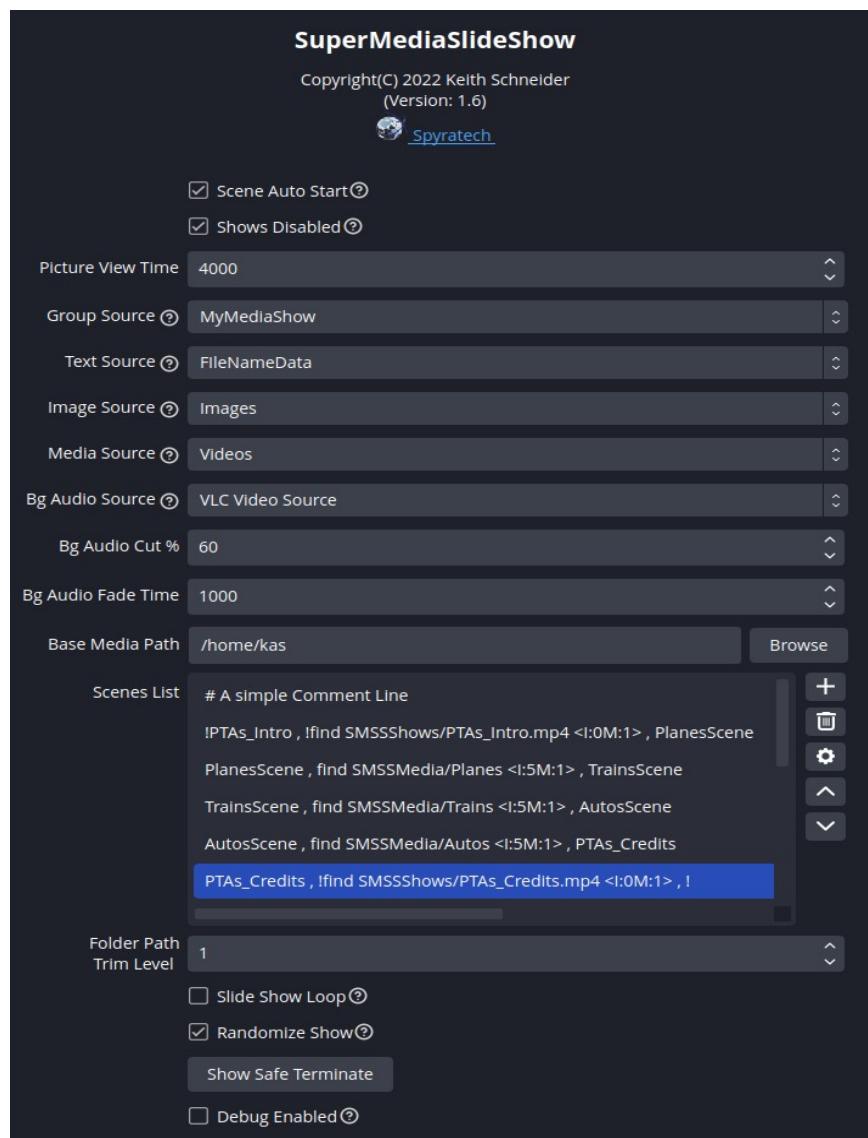
3. **Add an Image source.** Setup the size to the full screen. eg: Images
4. **Add a Media Source.** Setup the size to the full screen: eg: Videos
5. **Add a group to the scene and immediately mark it as deactivated.** Why? Because, depending on your setup progress and properties settings, once a Group becomes activated, the Slide Show might want to try to start. During setup, this would only get in the way and confuse things. eg: MyMediaShow
6. **Move each of the sources you added, into the Group and order them so the Text source is at the top.** The Text source should be at the top so it will overlay the Images and Videos sources so the text will be visible on top of any Image and or Video during the show. It should appear as follows:



7 Properties

The heart of the Control Panel of a slide show(s) is the settings in the SMSS Script Properties Window. <any of these settings are largely self explanatory, especially for even a somewhat experienced OBS-Studio user. However, a few of these settings may need some detailed explanation. First, let's take a look at a fully configured properties settings window.

Note – Selection Drop Down List fields that require you select sources from in the Properties/Settings Window are populated the *first* time the Scripts Window is opened. This is notable because if you have not configured any sources for a show (or have a fresh new install of OBS-Studio) and you enter the Scripts SMSS settings Window, these field selection boxes may be empty! Closing and re-opening the Scripts Window will not change this. You can then go add some Scenes, Sources etc and go back to the Scripts Window and find the same fields still empty (and or out of date). The currently accepted way to cause a refresh of the Window is to click the **C** (reload scripts) Icon.



Next, let's have a detailed look at an explanation of each setting parameter.

Scene Auto Start

Scene Auto Start - Enables/Disables “Show Auto Starting”. When a valid Scene is entered, internal to OBS-Studio, it becomes activated. In this discussion, a valid scene means that it is named as a show source in the Scenes List within the SMSS properties settings (see below). When such a valid scene activates, SMSS will act to start a show. When setting up and configuring your show, this gets in the way, because it starts, or because you may not have yet defined all the desired sources and things do not work as you want them to and so on. It’s kind of like having your car automatically start up and begin driving away while you are trying to work on it. Point being – it is good to just turn off automatic show starting until you are ready to let them run. This setting also inhibits the automatic switching of scenes at the end of a show even if a Next-Scene is called for. Depending on the “Shows Disabled” setting, you can still start a show manually within a single scene for testing.

Shows Disabled

Enables/Disables ALL shows from starting. While “Scene Auto Start” may be disabled, this can still be Enabled, which allows the starting of a show by clicking to the “show” state of the “Show Control Group” (which in turn activates all its group members) and a show begins to run. This can also occur by the user navigating the scenes in OBS-Studio and entering a scene that holds a SMSS Slide Show “Show Control Group” where the SMSS Group is already set to a “show” state – in which case, that show starts. Again, similar to the “Scene Auto Start” switch, it might seem like the car is driving away by itself. The moral of the story is that you ought leave this disabled until you are ready to let shows run, either as a single scene show or a Multi-Scene orchestration.

Picture View Time

This is the Time in milliseconds that a Image (picture) will display during a show. Because the Image source can have Show/Hide transitions configured for it, the SMSS program will lookup the Image show/hide times and factor those times into the actual Picture View Time. When the Picture View Time timer fires, the Image visibility is set to false causing the Image source to move to becoming deactivated. But to become deactivated, it has to complete any configured hide transition. Therefore, the actual timer setting used to show the Image takes the Picture View Time less the Hide Transition time to actually show (in practice) the Image for the total requested time. This method works well unless the user configures a Image Source “Show Transition time” that is greater than the “Picture View Time”.

Group Source

This is often referred to as the “Show Control Group”. It is the name of a OBS-Studio source group that is intended to contain the Text, Image and Media Sources. Internal to OBS-Studio, when a group is activated, its members likewise activate with it and inversely when deactivated. SMSS was built to operate upon this core premise.

Text Source

This is the Name of a Text source that the SMSS show will use at run time to display some amount of the file specification for the current slide. SMSS currently supports either the (internal Id's) "**text_gdipplus**" or "**text_ft2_source**" sources. This field can be blank but is generally advised to be present. You can enable/disable the actual use of this source during a show in a number of ways. You can disable this source via the show/hide icon/button within the Show Control Group, group members list. You might be able to position this source within the list such that it is underneath the show screen layers so it is never seen. Note that when the SMSS show runs, it checks the visibility status of the Text, Image and Media sources. Any of these that are set to disabled/invisible, will not be used during the show (regardless of the user Clicking the icon for them during a show). Media items that are collected from the "Media Collection Command" that match these disabled sources will be skipped/ignored. Note also that SMSS never manipulates the visibility of the Text Source during the show. It is not the focus of the show. It is sort of just included in the show as a ride-along to show the (all or a part of) file specification.

Image Source

This is the Name of the Image source in your group that will show each Image item/file that will be presented during the Slide Show. It is shown for a user defined Picture Duration Time and internally a timer is used to effect this time period. Specifically, SMSS supports only the (internal id) "**image_source**" source. A selection here is Not Required, but is advised. It can be set inactive within the Show Control Group if you do not want it to show during the Slide Show. Here also, the visibility setting of the source at the start of a show is honored throughout the show. This means that if the source is set disabled at the beginning of the show, it will not be used during the show and all media items collected by the Media Collection Command will be ignored. It is therefore possible to configure a show and set both the Image and Media sources to invisible and then run the show and observe a result of nothing happened. Observing the actual program debug log would show that the show starts and finds no suitable media files to show because they are all invisible. Then the debug log would show that the presentation comes to a normal end.

Media Source

This is the Name of of a "Media Source" in your group that will play each Media item/file items that will be presented during the Slide Show. Specifically, SMSS currently supports only the (internal id) "**ffmpeg_source**" source. An intuitive distinction of this source from the Image source is that because this is a Video source, it naturally displays on the screen for the natural duration of the playing video itself. Internally, SMSS detects the ending of the video by means of the OBS-Studio Signal "media ended". This signal triggers when the media item runs itself to the end or, if the user were to select the source (while playing) and Clicks the square End/Stop button in the player controls. A selection here is not required but is advised. It has the same caveat notes as mentioned in the definition and usage of the Image source (above). Show/Hide transition times for this source are not taken into consideration by the SMSS program.

Background Audio Source

This is the Name of an Audio Source that is configured somewhere in your total OBS-Studio setup. It does not have to be configured into your Show Control Group or your scene. In fact, if you are building a Multi-Scene show, you would want your Background (Bg) Audio Source to be configured somewhere globally to OBS-Studio so that as you transition from scene to scene, the audio does not get interrupted or restarted and so on.

Background Audio Cut

This is expressed as a Cut-To Percentage value. When a Background (Bg) Audio Source is defined, each time a Video is played, SMSS will fade the Bg Audio Source to the volume percentage of its starting volume level. The transition occurs over the Bg Audio Fade Time setting time period, in milliseconds. Note that you would normally set your overall volume prior to starting a show. When the show starts, SMSS will determine the volume level and calculate the fade volume as the percentage of the current set volume. Attempting to change the overall volume during the show has no effect. It uses the values determined at the start of the show.

Background Audio Fade Time

This sets the fade in/out time (ms) used when a Media source is playing and a Bg Audio Source is defined. A setting of a half second to 1.5 seconds is usually a good choice.

Base Folder Path

This is the base System File Path that locates your Media files. This is initially set to your platforms HOME Folder. On Windows, this is the path location defined in the Environment variable %HOMEPATH%. On both Linux and MacOS, this is the \$HOME environment variable. This is usually the folder just above a folder like Pictures, Documents, Music and Downloads etc.

Linux: \$HOME = /home/dave

Windows: %HOMEPATH% = C:\Users\dave

This enables all platforms to use Media Collection Command(s) to find Relative Path Files and this setting is intended to locate the folder, *containing* your media folder. In all other examples in this document, examples, portray the “Pictures” folder on all platforms, as residing as a first level folder under the users HOME path ie: /home/dave/Pictures, C:\Users\dave\Pictures.

A big advantage of Relative Path File paths is portability.

Usage Scenario: On your desktop PC, You have media in \$HOME/SMSSMedia. You copy your SMSSMedia Folder to a USB SSD for a Laptop running OBS-Studio. You configure OBS-Studio on the Laptop to be the same as the Desktop PC, but instead of copying the USB SSD data to the Laptop, you leave the Data on the USB SSD. You use this setting in OBS-Studio to locate the Device/Folder Path that *contains* the SMSSMedia Folder. When the SMSS Show runs, it will CWD to this Base Media Device/Path to collect the files using your original Relative File Path Command(s). You may however need to adjust your Folder Trim levels to adjust for the device/path location on the USB device.

Scenes List

This is the core place where you define Show Scene(s), Media Collection Command(s), and optionally Next-Scene(s). You can add as many entries as you wish. Use # comment lines to aid in organization, hold a testing or copy(s) of an entry. Valid entries are as follows: (blank spaces optional)

```
[#] [!]SceneName, [<flags>]MediaCollectionCommand [, [!]Next-Scene]
```

Items in [] brackets are optional. The '#' makes the entry an ignored comment. Both the **SceneName** and **MediaCollectionCommand** are required. If **Next-Scene** is defined, a transition to that Scene will occur at Show completion. Just like in programming this makes a simple forward Linked List. **SceneName** and the optional **Next-Scene must be Exact and case-sensitive**. Only the 1st of any duplicate Scenes will be used. There is no concept of a default Next-Scene. A valid MediaCollectionCommand is an OS command, that, when executed returns a plain list of Media file specifications. The ! Character attached to a SceneName or Next-Scene name is a **Recording Control Directive** (see below). Study these examples:

```
Linux:      Scene1 , find Pictures -name "*.jpg" , Scene2
Windows(rel): Scene1 , dir Pictures/*.jpg /B /S , Scene2
Windows(abs): Scene1 , dir C:\Users\Dave\Pictures/*.jpg /B /S , Scene2
```

Entries are checked for valid Scene Names and the command as non-blank. Errors are tagged at the start of an entry enclosed in '<err>' brackets. Each Media Collection Command is test executed and the results are tagged in <> brackets following each Command. The tag <I:12M:5> consists of Image & Media file counts returned. Non Image & Media files ignored. If the test returns zero files, the entry is tagged with <CmdErr>.

You can specify a 'default' SceneName entry. When a configured Scene is activated, the CurrentScene is looked up in this list and that Scene Media Command is used. If a Scene is not found, then the "default" entry is used. If a default entry is not defined, then no show occurs. The 'default' keyword is not case-sensitive and was intended for simple single scene shows.

When a scene is activated, SMSS will lookup the current scene name (at this writing, cannot be default) and if the names match, a show will start. See the SceneAutoStart setting (above).

Notes on TAGS – Tags are <> enclosed entries automatically added to your Scenes List Entries to show errors or Media Collection Command validation. These Tags are automatically removed/ignored by SMSS when a proper lookup occurs. They are informative to the user. They are there to help you know if your Scene, Command & Next-Scene entries are valid.

Examples (1st two show no errors and a collection of 8 Images & 0 Media, 3rd is <CmdErr>:

```
Lin: Scene1, find Pictures -name "*.jpg" -print <I:8M:0>, Scene2
Win: Scene1, DIR C:\Users\Dave\Pictures/*.jpg /B /S <I:8M:0>, Scene2
Win: <CmdErr> Scene1, DIR C:\Users\Dave\Pictures/*.jpg /B /S <I:0M:0>, Scene2
```

Note <CmdErr> caused by the misspelled folder name (**Picsures**) in the above example.

Scenes List Flags

Scenes List Flags is an optional, specially formatted field that, when used, prefixes a Media Collection Command for the purpose of providing certain “Settings Override” values to be applied to the show for the given scene.

These flags are specified as one or more two character fields followed by a : character followed by a value. The flags field must be enclosed in <> angle brackets. Each flag item:value pair are NOT separated with commas and no space characters should be included. Flags are not case sensitive.

Flags are as follows:

- VT:nnn – Specifies Image View Period in ms – Must be greater than zero.
- AF:nnn – Specifies Bg Audio Fade Time in ms – Integer value 0 or greater.
- AP:nnn – Specifies Bg Audio Cut To Percentage – Integer value 0 to 100.
- RN:T or F or 1 or 0 – True/False using T or F (1 or 0 allowed also).
- QT:T or F or 1 or 0 – True/False causes TEXT Source for show to be set invisible.

Examples:

- <VT:4000AF:2000AP:20RN:TQT:T> All options specified. VT: Picture View Time of 4 seconds, AF: Bg Audio Fade Time of 2 seconds, AP: Bg Audio Fade Cut To Percent of 20%, RN: Randomize Show set True, QT: Quiet Text Mode – sets Text source to invisible.
- <QT:T> or <QT:1> Set the Text source to be invisible so the text (media file path specifications) are not seen on the screen. Note that <QT:F> or <QT:0> is the default.
- <RN:T> or <RN:1> Set the show to have its Media File(s) randomized.
- <AF:0AP:100> Effectively turns off Bg Audio Fading.

In practice, when you start stringing multiple shows together, you may find that you might like to alter the specific settings for each show. This is the means to do that. For example, When I made a family home photo video, I setup both Intro and Ending Scenes where I did not want the Text media file path specification to be visible. But for most all the other scenes, I wanted the media file specification paths to be shown.

I also setup my Intro and Ending scenes with Videos where I expressly wanted the Background Audio to fade to zero. I also had cases where some/all videos (recorded from old Super-8/16mm B/W Family Movies) in a show did not have a sound track and I therefore did not want the normal Background Audio Fade that I would want for a Media Video with a sound track.

The idea of these Flags is to allow a greater granularity of settings from show to show. Your choice.

Note: You might say that these settings would be more readable if separated with the “,” character and this is true. However, the whole SceneList entry thing is getting rather long as is and these flags, fields parse just fine by the lua pattern matching tools and so on...

Recording Control Directives – The ! Character

There are a few rules to When and Where this character is used but when understood, it makes intuitive sense.

1. When the ! Is prefixed to a SceneName part of an entry, it generally means that Recording should start. With a few caveats of course:

1. **If recording is started** then any Bg Audio volume is immediately set to Zero followed by a Fade-Up of the Bg Audio Source volume to its normal level. This facilitates the smooth opening of a new Recorded Video.
2. **If Recording was already active**, then neither the Recording Start or Volume to Zero/Fade Up will occur. This works this way to allow seamless flow in an existing recording.

Note: If multiple entries exist to form a Multi-Scene SMSS Show, then, theoretically and in practice, (although arguably illogical) each entry could have their SceneName prefixed with the ! Character and they should still all flow together as a singular movie.

2. When the ! Character is attached to a **Next-Scene** part of an entry, **or stands by itself**, this means two different things.

1. **When prefixing a valid Next-Scene**, at the end of that entries show, the Bg Audio is faded to zero and then SMSS will switch to Next-Scene.
2. **When ! stands alone** in the Next-Scene field position, it is taken to mean that the Bg Audio will fade to Zero and then Recording will stop.

Together, these behaviors can be taken advantage of to construct:

1. A full Show Video from Recording Start to Recording Stop with Bg Audio fading in at the start and fading out at the end.
2. A segment of a larger recording where recording (again, if already in-progress) is not interrupted nor Bg Audio altered. The SMSS Scene(s) are shown through their ordered Linked-List sequence until finished and transitions into a non-SMSS segment of a recording.
3. And more. It can facilitate an SMSS show transitioning into or from non SMSS segment(s) of a show overall. You can use these controls to achieve your desired effect.

Note that when generating a Video that has auto recording start/stop (via !) with a Background Video. The fade in/out time used is double the specified Background Audio Source fade in/out time.

Folder Path Trim Level

This specifies how many levels of folder names should be stripped from the displayed items total file specification. Often the top most file folder names are not useful to the show itself. This allows them to be removed from the displayed file name and yet keep some number of folder levels that may add context to the current media item shown.

You probably do not want your displayed Images text to appear as:

C:\Users\Dave\Pictures\Moms_2006_90th_Birthday/image21332.jpg

When you can trim the left 4 levels to get a nice displayed text as:

Moms_2006_90th_Birthday/image21332.jpg

Slide Show Loop

This setting works as it states. It makes a show loop forever. Note that setting this while attempting to run a Multi-Scene Show will never switch Scenes as the first show sequence will loop forever and never finish.

Randomize Show

This setting simply randomizes the Media Collection List items within a show.

The Show Safe Terminate Button

This button is the same as the elsewhere described HotKey. It is important to know that if you use this Hotkey/Button, it DOES NOT stop any recording initiated by SMSS nor does it change to a Next-Scene and so on. It merely stops the show from progressing.

Debug Enabled

This setting enables debug mode. You can open the script log window and see debug output therein. When Debug Mode is enabled, additional property/settings fields will appear that allow you to select various debug levels, which shows more and more detailed messages during the execution of a Slide Show. You can increase and decrease these levels on-the-fly during a show. For the curious user of SMSS, it may be a bit entertaining to watch this output while a show is running and please understand that increasing to high debug levels adds more load to your system at the same time. I left all the developer debug code in the program for release and thought it might be interesting to some users but if you venture into this area, you are on your own. There could also be a case where an issue might be recorded in such a log and useful to provide to the developer for debugging/problem identification. Note also that the script log is the place to view this output on Windows but if you are on Linux (MacOS, probably same), and you need/want to take a serious look at this output, then open a terminal command window and start OBS-Studio from the command line. You'll be able to see all this output, on-the-fly. Lastly, this logging output is also captured in the run log files that OBS-Studio captures. These log files are useful for developer feedback when problems occur.

8 Multi-Scene Example

For illustrative purposes, I have shown below a set of “Scenes List” items (see Properties) that show what drives and encompasses a Multi-Scene show.

Largely a Multi-Scene show means that you have created multiple Scenes in OBS-Studio and in some form or another added into each, another instance of the same Show Control Group you created in the “Organizing Sources” section of this document. You have thought through the flow of your Multi-Scene presentation and planned accordingly.

The core of the show (aside from the setup/config of all the Scenes and sources et al) is the Scenes List entries in the SMSS Properties Window. Below is an example I thought of for Demo Purposes. I spaced out (but still valid) the entries to improve overall readability:

```
!PTAs_Intro ,<QT:T>find SMSSShows/PTAs_Intro.mp4      , PlanesScene
PlanesScene ,           find SMSSMedia/Planes          , TrainsScene
TrainsScene ,           find SMSSMedia/Trains         , AutosScene
AutosScene ,           find SMSSMedia/Autos          , PTAs_Credits
PTAs_Credits,<QT:T>find SMSSShows/PTAs_Credits.mp4 , !
```

You can easily see that each entry specifies (finds) some number of Media files to be shown within the defined scene. When the show completes, SMSS will switch to the Next-Scene in a sequence of chained shows. Think “Linked List”. Lastly, the PTAs_Credits is entered and the show ends.

But did you catch the first and last entries? Look closer. Note in the PTAs_Intro Scene, I specify a Media Collection Command that finds a single Media mp4 file. I used <QT:T> To inhibit the visibility of the Text source for that scene. Suppose that that Video was myself (gloriously of course) doing a Video Intro to the show. When that Video completes, it automatically switches to the Planes Scene and the show continues through each Scene-Show until it reaches the PTAs_Credits scene where it is set to find another singular Video file that may be a video of a vertical scrolling text source listing credits. Here in the Credits I also used the <QT:T> to make the text source invisible.

Note also the first and last entries are also using the ! character attached to the SceneName and Next-Scene. Here, the first entry is telling the program to start recording on the opening PTAs_Intro scene and the last entry Next-Scene is empty except for the lone ! Character. This tells SMSS to fade down any BgAudio source to zero and then stop recording without moving to a next scene because there is nothing to switch to. It is also important to know that when a SceneName entry calls out the ! Character, it also implies that any Bg Audio source be immediately set to Zero Volume followed by a Recording Start and then a Volume Fade Up to normal. But there is more! If at Scene Startup, Recording is already active, neither a recording start or BgAudio fade up occurs. This allows for seamless continuity if the SMSS scene collection starts in a situation where some other OBS-Studio scenes and a recording effort is already in progress.

On top of all this, I might have also setup a base Scene (that I added to each of the above listed Scenes) where I setup some screen background(s). And since each show may include some Videos, I could add into the base Scene a audio source configured to be a Bg Audio Source that SMSS will fade in/out when those Videos appear in the course of the overall show execution.

9 Support Services

Support services are simply based on posted questions in the OBS-Studio forums and possibly obtaining direct email dialog with the Author.

Use the forum to post the good and the bad about this software. We can handle it :-).

Have any ideas for updates? Post them in the forum also.

10 Miscellaneous

Media Collection Commands – Why?

The reason I chose to defer the collection of files for the presentation to an OS command, is, IMHO, that I could easily read from a pipe and push that burden onto the user. A *nix oriented person might like this. You can go as easy as prepare your own text file of file specifications (one per line) and then specify a simple OS command (Windows: type) (Linux: cat) whatever to list the file contents and that ought to work. If you want to just drill into a single folder path, you can use a Linux find command or on Windows use a DIR command as needed (examples provided elsewhere in this document). Works well as far as I see it...

I did not want to recurs into folders with lua as it appears that the core language tools needed to enable an effective drill into folder structures is lacking. Perhaps some comprehensive file tools will appear in the language or some libraries of tools will show up (if not already somewhere that I know not about). Compare with Perl, PHP, Python and many others, “base” Lua loses the challenge in a heartbeat. Anyway, I consider the Linux find command to work well and think that the Windows DIR command is adequate (despite it always returning absolute paths). And if you have your media scattered all over your folder hierarchy on your system, then make a script that compiles the list to your needs and specify that in your Media Collection command. Or, because we have such big disks these days, make some consolidated and organized copies of your media into a Slide Show Folder structure.

Current Working Directory – Thoughts

Linux and MacOS use the concept of the HOME folder entirely. OBS-Studio is installed in OS system folders (like Windows) but when OBS-Studio runs, it runs with (what is called) the CD (sometimes called CWD – Current Working Directory) set to the HOME folder. This allows most all user actions and paradigms to use Relative File Paths naturally. Windows platforms have never 100% embraced this integral to the natural use of a Windows PC. Also, because of the way that apps are installed and launched, they often do not start with the concept of its (cd) current directory set to the users home folder. This is why the notion of (especially in the early days of Windows) the question of -- Where do I store my files? -- has mostly been a point of confusion, nay, contention. Later in its life, Windows was improved it but it's never been a total clean/clear implementation.

This explains why on Windows I see other OBS-Studio forum users asking for the ability to use relative file specifications. You largely cannot, [Relative to What!] and thus are forced to use full path specifications. Even OBS-Studio itself is installed (it appears this way to me) such that it depends on its CD being the its install folder. Try changing the OBS-Studio Desktop Icon properties to set the start directory to be your home folder. Then run OBS-Studio. It breaks trying to find ini Locale files. Because of this dependency, a relative path Media Collection Command acts to search for files relative to the OBS-Studio install path, not

your HOME path.

However, with the addition of the SMSS “Base Folder Path” setting, you can use Relative oriented find/search commands and incorporate them into your “Scenes List” settings. Relative to What? The Base Folder Path, of course!

Source Transitions – Thoughts

On the subject of **Image Source Transitions** -- At this time, the rotation of images is based on merely doing a show/hide (via sceneitem visibility) of a single image source.

Consider the difference observed and implemented in the OBS-Studio built-in Image Slide Show source. It transitions within itself from two Images with a single seamless A to B transition.

The current implementation used by the SMSS program is that it switches between each image in multiple steps. Image A (source) enters into a hide transition (and deactivates) to whatever the current screen background is. Then that Image source is loaded with a new file specification and activated. The source begins its show transition until complete and the Image remains for some period of time.

The cycle then continues. Thus, it's surely helpful to have your media files on fast backing store (ie: SSD) and a fast computer to speed this cycle up as best as possible.

Perhaps in a later update multiple image sources can be setup to fast parallel/asynchronously show/hide to near eliminate Image load latency and as the OBS-Studio Lua API gets more understood and mature, along with Multiple Image sources the transitions can be managed more perfectly...

Futures

I've been seriously thinking of, and already testing out making this script into a registered source. This way, each instance of the source (logically one per scene, but...) ought have it's own settings, media collection command etc. Thus, the notion of the ScenesList (and Flags) as used in this implementation would be gone and yet the linkage from scene to scene with fully individualized settings per scene is retained. The script implementation (herein) has you setup multiple scenes and optionally add each show settings to attempt to match such a functionality. The settings Window in OBS (at this time) divides the page into two sides vertically and this wastes precious UI space for the script properties settings configuration. I'm also thinking that if implemented as a source, the race conditions that occur in the existing script implementation are likely reduced and more manageable. If the “Shows Disabled” setting is not checked and you start bouncing around the scenes while working on a significant setup of scenes, the number of scenes and sources activation's/deactivation's can start some real thrashing. Hence why I recommend that this setting be used often while working on a show configuration.

Be Creative!

Choose Wisely, Choose Liberty!

Do Good in the World, Remember God and that God Wins!