1. Summary

MRLControl provides a graphical user interface for managing FFMPeg (or VLC or other) command line scripts. It provides for building/maintain complex commands modifiable by user inputs. It extends the FFMPeg capabilities by allowing user selection of parameters and multiple parameters in an FFMPeg input file. It manages FFMPeg inputs by using scripts, playlists (Layouts) and User Strings. Scripts are one or more command line entries. Playlists are a set of scripts that can be displayed sequentially of merged into a single screen(Layout). User Strings are either System or user-defined data that can be combined to build either other user strings or scripts.

Major Features include

- Display one or more input sources (Files or Stream)
- Record either input sources (file or stream) , Desktop or selected windows
- Stream either inputs or displayed windows
- User selection of parameters via list of dialogue
- Storage/reuse of portions of command
- Access to system parameters (e.g., Microphone, Cameras, etc.

2. MRL Software

MRL Software was formed in 2016 to develop software to fully employ the significant MSOffice capabilities while Simplifying the User Knowledge Required to Use their products. In addition to simplifying MSOffice usage, our goals are to Eliminate Data Base Administration Complexity, Office "Development" Complexity to better Deliver Effective Software products. MRL Software has Expertise in Software Development (40+ Years), Windows (20+ Years), and MS Office Object Models (10+ Years). MRLControl is the first Video product, first Open Source Product, and first Visual studio product. Other products include 1) DataManager- an integrated development environment Integrated with Microsoft Office Environment (as an Excel Add-in) to facilitate MSOffice Applications development, a personal Library Manager, and a tool for developing Windows Help. We are actively seeking Vertically integrated Product partnering opportunities.

3. Quick Start Guide

This step discusses how to install the download files, configure the DataManager Excel Add-In, customize the screen, and understand the input formats for loading your data.

a. Installing the Files

First install FFMPeg and optionally VLC from the source website. To use VLC, you must add the directory to your path statement. Also, you should configure the program for minimal interface (and possible to stop after closing the command window. Then download the MRLControl zip file are available this web site. These consist of the

MRLControl.exe, MRL DLL's and several tool DLL's available as open source. Unzip the files to a directory of your choice. The software contains four subdirectories 1) config has all configuration files, 2) media has sample input files, 2) mediaout is user for output from the sample scripts, and 4) tutorials contain user videos and other help information. Detailed information on the sample scripts is available in the Samples section. Data file formats (all text) and command debugging approaches are discussed in the advanced user section

b. Sample Scripts

This section provides a description of the sample scripts and intended purpose

Play Scripts	**Purpose**	**Comments**
FFShowVideoFile	Demo-ffplay by displaying user selected video file	User Selects video file that is displayed. Position id stored in script and can be updated by dragging(and saving).
FFShowVideoFileFilters	Demo-ffplay by displaying user selected video file with FFMPeg filter	User Selects video file that is displayed with selected filter. Position id stored in script and can be updated by dragging(and saving).
VLCShowVideoFile	Demo-VLC by displaying user selected video file	User Selects video file that is displayed. Position id stored in script and can be updated by dragging(and saving).
VLCShowVideoFileFilters	Demo-VLC by displaying user selected video file with filter	User Selects video file that is displayed with selected filter. Position id stored in script and can be updated by dragging(and saving).
VLCShowImageFile	Demo-VLC by displaying user selected image file	User Selects image file that is displayed. Position id stored in script and can be updated by dragging(and saving).
DefShowVideoFile	Show video file with default video program	User Selects video file that is displayed. Position id stored in script and can be updated by dragging(and saving).
DefShowImageFile	Show image file with default image program	User Selects image file that is displayed. Position id stored in script and can be updated by dragging(and saving).
DisplayCamerawithMike	Displays Camera with Mike input	User to camera and mike (if needed)
PLayDirectory	Randomly play files from selected directory	Uses VLC random capability. Note VLC must not be set to Quit after playing input.
DisplayLANCamera	Shows displaying Cellphone camera	Displays stream from Android IP Webcam. Note hardcoded local iP address

Process (Batch) Scripts	**Purpose**	**Comments**
ConvertAVIMPG	Convert file from AVI to MPD	Allows user to select files individually or from a directory and run script to convert each selected file.

RecordCamerawithMike	Shows recording Camera and Microphone	User Starts and stops recording from Process Window
MergeTwoVideos	Combines two videos into a side by side video	User selects two video files separately and then runs script to create output
MergeFiles4	Combines four videos into an output file	User selects two video files as multiple files and then runs script to create output
RecordDesktop	Records desktop	User starts and stops of desktop using process dialogue.
RecordWindow	Records User Selected Displayed Window	User Selects window to be recorded and starts script
MergeClips	Demos expanded file processing by merging clips specified in input file with name start and number of frames	Script uses MergeCLips.txt. e]to merge clips from two separate video files. Starts command and produces output video.
MergeClips1	Demos expanded file processing by merging clips specified in input file with name start and number of frames	Script uses MergeCLips1.txt. e]to merge multiple clips from same video file. Starts command and produces output video.
PlayFile	Plays user selected files	User selects files or directory from Process dialogue then displays them sequentially.
OverlayTitle	Overlay image at user specified position between two times and create output video	Demos user input dialogue(see user string Overlay Image) to Overlay a title image on video at user specified position, start and stop times.
interleave	Uses FFMPeg interleaving	Creates output video interleaving frames from two video files. Files must be compatible.
Output Scripts	**Purpose**	**Comments**
RecordInputMike	Demo recording camera and mike	Records a Camera/File
StreamWindow	Stream the video source window without audio	Stream the video source window without audio. Must set up User Strings for STREAMTARGET$ and $UTUBEKEY$
RecordWAudio	Records Window and its audio	Stream the video source window with audio.
StreamWAudio	Stream the video source window with audio	Stream the video source window with audio. Must set up User Strings for STREAMTARGET$ and $UTUBEKEY$
RecordAudio	Records Audio Output from input	User Selects input file.
StreamAudio	Streams Audio Output from input	User Selects input file. Must set up User Strings for STREAMTARGET$ and $UTUBEKEY$
RecordMike	Records Audio Output from Microphone	
StreamMike	Streams Audio Output from Microphone	Must set up User Strings for STREAMTARGET$ and $UTUBEKEY$

RecordWMike	Records Window and Audio Output from Microphone	
StreamWMike	Streams Window and Audio Output from Microphone	Must set up User Strings for STREAMTARGET$ and $UTUBEKEY$
RecordLayout	Records Desktop	Note: Cam be used as parameter on MRLControl.exe command shortcut to record session.
VoiceOver	Records Desktop with voiceover	Note: Cam be used as parameter on MRLControl.exe command shortcut to record session.

4. Basic Data Concepts

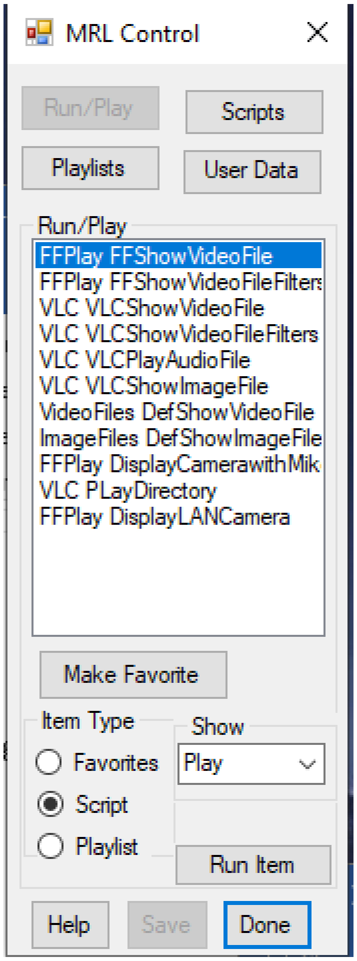
MRLControl utilizes an object model all based on developing textual data strings to be processed in a batch command window. The components are Scripts, Playlists (Layouts) and Strings (either a system ProSign or User String.

- A Script is one or more batch commands executed as a windows process running as a background task. There are three types of scripts 1) Play scripts are designed to run the script and display a window 2) Process scripts as intended to bring up a "batch" window to allow the user to select files, directories or input file then execute the script(or execute scripts that have no output to screen), output scripts are used to record or stream output from the screen/window.
- Playlist is one or more scripts that can be run either sequentially (Playlist0 or displayed concurrently in a single window (Layout). ProSigns are system key words (e. g., MEDIADIRECTORY to include the default subdirectory in a command.
- User strings are the same except they are build by the user. For example, the user may select from a set of FFMPeg filters whose syntax (including optional input parameters) defined by a set of User strings.

b. Screen/Parameter Setup

The Start-up screens as shown below changes to support playing Scripts or Layouts and Editing Scripts, Playlists or User Strings. All screens have four buttons at the top to select function (Run/Play, Scripts, Playlists, and User Data to invoke the functions. They have three buttons at the bottom for Help, Save and Done. The Save button is enabled if any data has been updated (Including user selections and Screen Position. The user may place this window anywhere on the screen by dragging and then saving the new position.

- ### Playing Scripts or Playlists/Layouts

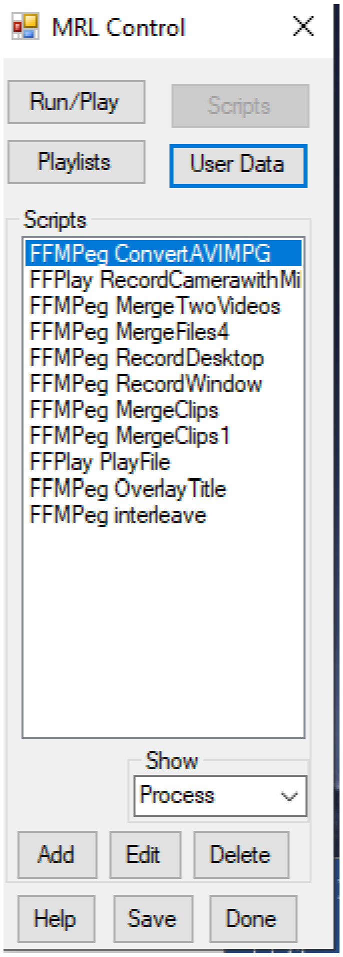


Item Type: ○ Favorites ● Script ○ Playlist

Show: Play

This screen shows up when the Run/Play button is selected. It provides the capability to run items. The list box shows the items to run. Items displayed depend on selection of the item type. Favorites are those selected by the used as heavy usage. This is done by the Make Favorite button. Script displays available scripts. These may be partitioned by the Show Combo box into All, Play and Process. Playlists displays the available playlists. If the selected playlist has been made into a layout, a Run Layout will be displayed. Clicking the Run Item starts the script/playlist. Clicking the Run Layout starts a layout window. If a script is a process script, the Batch process dialogue will appear, otherwise the output will be displayed. Note an output window location can be changes by dragging and saving the new position.

- ## Editing Scripts



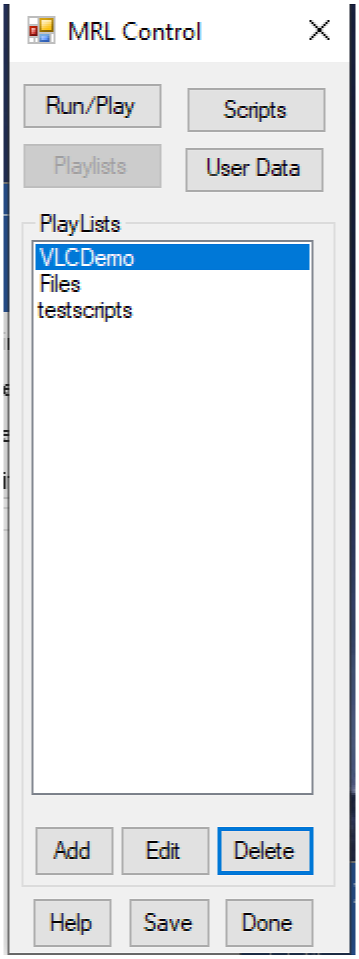
Scripts

FFMPeg ConvertAVIMPG
FFPlay RecordCamerawithMi
FFMPeg MergeTwoVideos
FFMPeg MergeFiles4
FFMPeg RecordDesktop
FFMPeg RecordWindow
FFMPeg MergeClips
FFMPeg MergeClips1
FFPlay PlayFile
FFMPeg OverlayTitle
FFMPeg interleave

Show: Process

This screen shows up when the Scripts button is selected. It provides the capability to edit script items. The list box shows the items to process. Items displayed depend on selection of the show type. Values are All, Play, Process and Output. The Add and Edit buttons activate a Dialogue to create/Edit a script item. See below for the details of updating a script. The Delete button deletes the item after confirmation. Note changes are not finalized until Save button is clicked.

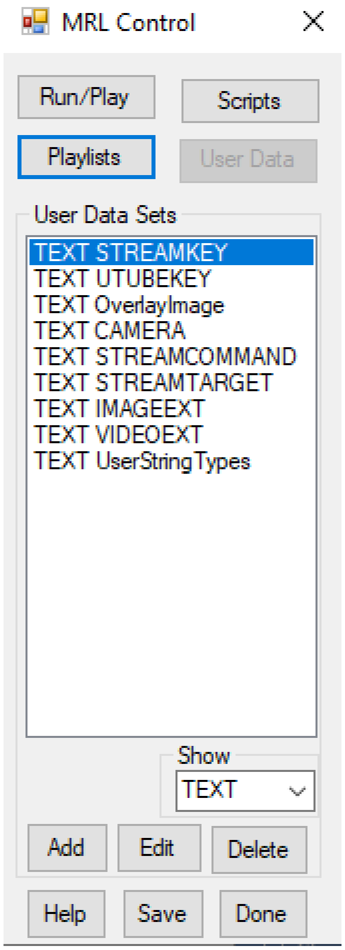
- ### Editing Playlists/Layouts



VLCDemo
Files
testscripts

This screen shows up when the Scripts button is selected. It provides the capability to edit playlist/Layout items. The list box shows the items to process. The Add and Edit buttons activate a Dialogue to Create/Edit a Playlist/Layout item. See below for the details of updating a Playlist/Layout. The Delete button deletes the item after confirmation. Note changes are not finalized until Save button is clicked.

• Editing User Strings



User Data Sets

TEXT STREAMKEY
TEXT UTUBEKEY
TEXT OverlayImage
TEXT CAMERA
TEXT STREAMCOMMAND
TEXT STREAMTARGET
TEXT IMAGEEXT
TEXT VIDEOEXT
TEXT UserStringTypes

Show: TEXT

This screen shows up when the User Data button is selected. It provides the capability to edit User String items. The list box shows the items to process. The items displayed depend on the Show type selected. Types include ALL, USER, TEXT and user types. The Add and Edit buttons activate a Dialogue to Create/Edit a User String item. See below for the details of updating a User String. The Delete button deletes the item after confirmation. Note changes are not finalized until Save button is clicked.

c. Using ProSigns and User Strings

System ProSigns and User Strings are a key to implementing a flexible set of FFMPeg command lines. They allow the user to partition the command and provide for user input, A simple example of using these is:

ffplay -i "$FILE\VIDEOEXT$" $FFMPEGFilters$, which allows the user to select and video file using one of selected FFMPeg filters. The parts of this command are ProSigns and User Strings (Enclosed by $):

- FILE – This is a ProSign that allows the user to select an input file using the File Dialogue
- VIDEOEXT – This is a User String containing the file extension to be passed to the file Dialogue.
- FFMPEGFilters – This is a user type with several User String items (FFMPeg filter strings) that the user can select from.

Using these capabilities allow a user to display any video input file and apply one of several FFMPeg filters to it. Adding a different filter requires adding a User String item. The following table defines the System ProSigns. For each, if there are multiple options, a List is displayed to user. If only one option (e., g. Computer Camera), that is user.

Function	**Syntax**	**Comments**
Get Local Cameras/Video Inputs	$LOCALVIDEODEVICE$	Selects Computer (or USB) input video deice
Get Local Cameras/Video Inputs	$LOCALAUDIODEVICE$	Selects Microphone Input.
Get Filename without directory	$FILEONLY$	Gets only Filename
Get Multiple Filenames	$FILES$	Gets multiple filenames separated by commas. Directory appended if not default media directory
Use Input files Directory	$MEDIADIRECTORY$	Includes the default media directory in command. Note uses current Directory with Media sub-directors. If a separate path, set up a User String instead. No User Input.
Use Output files Directory	$MEDIAOUTDIRECTORY$	Includes the default media directory in command. Note uses current Directory with Mediaout sub-directors. If a separate path, set up a User String instead. No User Input.
Select User Window	$USERWINDOW$	Allows the user to select a displayed window. Generally, user for Output scripts.
Select display Window	$WINDOW$	User to select the display window of video output. Generally, user for Output scripts.
Get Size of Display Window	$SIZE$	Use the current size HHxWW (Height and Width) of the display window
Get the current directory	$DIRECTORY$	Includes the default media directory in command.
Select a Local Device IP address	$LANADDRESS$	Allows User to select an IP address on local area network.
Provide for processing of parameters from an input file	$PROCESSXXX$	User to pass parameters from an input file to command. XXX is a column heading whose value will be inserted into command. First row of file is Names, and remaining row is executed. File identifies the input file, and

		Outfile will generate a new file with FILE name, item number and, and extension. Used to copy segments of a file to merge into a single output file.
Create a unique output file name.	$NEWFILExxx.ext$	Creates a unique output name. format is XXX(Date/Time).EXT.

d. Input Data and Processes

e. Advanced Topics