



USER MANUAL

VERSION 1.0



The MotaVision Video Controller is a powerful platform for controlling simultaneous video playback across multiple video display screens. It is a 64-Bit Native Windows application that is tightly integrated with the low level Windows environment.

The integrated help topics and the application in general assume the user has some experience and familiarity with basic Windows configuration and usage practices. The setup of network adapters, IP addresses, and the custom layout and arrangement of multiple displays are handled in the standard Windows Settings application.

Using the Help System

The integrated help system can be activated by selecting **View Help** from the application **Help** menu. The standard F1 help key is also available when working with any of the main application tabs. When the F1 key is pressed, the help viewer will open to the most relevant topic depending on the current activity.

If you read nothing else, we strongly recommend you review the following help page. It provides critical information information that will help you get started quickly.

Important Information



Important Information

Application Icons

During installation, icons were created on the Windows Desktop and in the Start Menu for launching the application in different ways. All of the core application functionality (video playback, script execution, etc.) is active no matter which icon is used to start the application.

MotaVision Setup Mode	This icon is used to launch the application in setup mode. The main application window is visible and all of the features required to configure and manage the system for automatic playback or remote control are available.
MotaVision Kiosk Mode	This icon is used to launch the application in kiosk mode. The main application window and mouse cursor are completely hidden to ensure only the video windows are visible. In addition, features that can only be managed in Setup Mode will have the corresponding menu entry disabled.

Application Hotkeys

The table below lists hotkeys that are available whenever the MotaVision Video Controller is running.

CTRL + ALT + BACKSPACE	This hotkey will toggle the main user interface between hidden and visible.
CTRL + ALT + HOME	This hotkey will force the main application window to be visible and move it to the primary display.

Web Browser Access

To connect to the MotaVision web interface from a remote computer, just enter the IP address of the MotaVision computer in the URL or search bar of the browser. In rare cases, you may have to explicitly enter `http://` (**NOT** `https://`) followed by the IP address if the browser has trouble connecting. Here's an example:

`http://192.168.10.123`

Remember to replace 192.168.10.123 in the example with the actual IP address configured on the MotaVision system.

Importing Media Files

There are several ways to import video, image, and script files into the MotaVision system.

Removable Drive	When you plug a USB thumb drive or other removable drive into the computer, by default MotaVision will open the multimedia import dialog. This dialog will show a list of all compatible media files on the device along with a checkbox next to each file, allowing selection of exactly which files will be imported. Each connected drive will also appear with an Import Drive ... entry in the Media menu for easy access to the import feature.
Web Browser	When you access web browser version of the MotaVision user interface, an additional Upload Content button will appear next to the other buttons in the System Controls section. Clicking this button will open the MotaVision Content Upload page where you can drag-and-drop or click-and-select files on the remote computer for uploading to the MotaVision system.
Network Connection	When Network Sharing is enabled from the Media menu, remote computers can connect to the MotaVision system using the configured credentials. The system creates shares for the Video, Image, and Script file directories. Once connected, files can be copied from the remote computer to the appropriate MotaVision share directory. On Windows, the File Explorer is typically used to connect to shares on remote computers. On MacOS there is generally a <i>Connect to Server...</i> option inside one of the Finder menus. For more technical details on connecting to the shared network folders, you may need to consult the documentation for the operating system running on the source computer.
Windows Explorer	If you are comfortable and experienced with Windows, you can use File Explorer or even the command line to copy files from any available source directly to the MotaVision media directories. Check the Configuration dialog from the File menu to see the directory location where the system stores media files of each supported type.



Graphical User Interface

The MotaVision system provides two separate but very similar user interfaces for controlling and managing the MotaVision system.

Native Windows Interface

The primary user interface is the native Windows interface. It runs directly on the Windows desktop and is tightly integrated with the Windows environment and the underlying media control systems. This provides a more fluid and responsive experience, and allows the different application tabs to have a rich set of features that simplifies configuration and management of the system. It is also the only interface where low level system configuration can be managed.

Web Browser Interface

The web browser interface has layout and functionality very similar to the native Windows interface. However, because it runs remotely across the network, the system controls and display updates in the browser may not feel as fast or fluid. It also does not support managing the low level system configuration.

To connect to the MotaVision web interface from a remote computer, just enter the IP address of the MotaVision computer in the URL or search bar of the browser. In rare cases, you may have to explicitly enter `http://` (**NOT** `https://`) followed by the IP address if the browser has trouble connecting. Here's an example:

`http://192.168.10.123`

Remember to replace 192.168.10.123 in the example with the actual IP address configured on the MotaVision system.



Screen Selection

Used to select which video screen is targeted by the controls on the Video Control, Overlay Image, and Overlay Text tabs.

System Controls

- **Stop All Screens** - Click to stop video playback on all screens
- **Clear All Overlays** - Click to clear all image and text overlays on all screens
- **Stop All Scripts** - Click to stop all active scripts

MIDI Clock

Displays timecode received on the network. When network timecode is detected, the decoded HH:MM:SS.FF value will be displayed. When no timecode is detected, the value 00:00:00.00 will be displayed. On the System Control tab, the MIDI Triggers table can be used to configure scripts to be run at specific timecode offsets. This allows video playback and control that is synchronized to an external timecode source.



Show Log Window

Toggles visibility of the MotaVision application log window. In addition to error and warning messages, the log displays informational and optional debug messages that can be helpful in identifying and diagnosing problems with media files, control scripts, and other types of issues.

Network Security

Opens the network security dialog where access to network features can be controlled. When the **Allow List** is empty, UDP commands and browser HTTP commands are permitted from any computer on the network.

For maximum security, the IP address of trusted clients should be added to the **Allow List** to ensure network traffic from all other network sources is blocked.

Application Font

Opens the font selection dialog where you can change the display font used for drawing the user interface.

AutoRun Setup (Setup Mode Only)

Opens the AutoRun dialog used to configure MotaVision to launch automatically on computer startup.

Kiosk Mode AutoRun Script

Used to select an optional script that will run automatically when MotaVision launches in Kiosk Mode. Selecting *Script AutoRun Disabled* will disable the mechanism which is common in situations where the system is being controlled from another device via UDP commands.

MotaVision AutoRun

This checkbox is used to enable the Windows feature for launching the MotaVision application automatically when a user logs into the computer.

Windows AutoLogon

This section is used to enable the Windows AutoLogon feature. When the checkbox is enabled and a valid Windows Username and Password are provided, the Windows logon screen will be skipped and the configured user will be logged into the system automatically.

Configuration

Script/Video/Image Files Path

These fields can be used to change the disk location where script, video, and image files are stored. The values should normally not be changed from default unless you have a full understanding of filesystem operations and have a specific reason to modify them.

Daily Reboot Time and checkbox

The best written software is highly dependent on the operating system upon which it executes. If that operating system doesn't always play nice, it can be useful to reboot the computer on occasion to keep things running smoothly. These controls allow enabling daily reboot and setting a time when the brief interruption from a system reboot will have the least amount of impact.

Network Port and Network Screen

MotaVision can be controlled via text commands sent to the UDP ports configured here. For compatibility with other video playback systems, two separate network ports are available. Commands sent to these ports will by default apply to the screen associated with the port in this section. Network commands that include a screen= parameter will always apply to the screen specified in the command and can be sent to either port.

MIDI Timecode Port and Multicast Address

MotaVision listens on the network for MIDI Timecode broadcast to the UDP port configured in this section. This allows the system to trigger scripts that run at specific timecode markers. In some cases, the timecode source source may transmit to a multicast address. Receiving timecode generated in this way requires MotaVision to subscribe to the target multicast address which can be entered here.

Warn before overwriting any existing file

Enable to display a confirmation dialog whenever a file save operation will overwrite an existing file with the same name.

Warn before deleting any existing file

Enable to display a confirmation dialog whenever a file delete operation is performed.

Move deleted files to the recycle bin

Enable to move deleted files to the Windows Recycle Bin instead of deleting them immediately. This allows the files to be restored from the Recycle Bin if needed at a later time. When this option is enabled, you may need to periodically review and empty the Recycle Bin to prevent disk space from being exhausted.

Import popup on media insertion

Enable to have the media import dialog automatically displayed when a USB stick or other removable drive is inserted. When disabled, the media import dialog can be activated manually from the corresponding **"Import Drive..."** entry on the Media menu.

Enable debug messages in the Log Window

Enable to include low level debug messages in the application log window.

Enable Log Window network broadcast

Used to enable the broadcast of all application log messages on the network. This is an advanced feature that requires special software and technical knowlege, and should not normally be enabled unless requested by customer support.

Enable Log Window capture file

Used to enable the capture of all application log messages to a daily log file. This feature is disabled by default to prevent a large number of log files from accumulating and filling up the disk over time.

Exit

Used to shutdown and exit the MotaVision application completely.

IMPORTANT NOTE:

The standard X or close icon in the top right corner of the main application window does not exit the application. It simply closes or hides the main application window so it does not obstruct video playback. This menu option is the only way to close and exit the application.



Network Sharing (Setup Mode Only)

Opens the **Network Sharing** dialog where the user credentials and enable checkbox can be set to allow connections from remote computers to the MotaVision system. When enabled, the system creates network shares for the Video, Image, and Script file directories. Remote users can connect to these network shares using the configured credentials, allowing media files to be copied directly from a remote computer to the appropriate MotaVision directory.

On Windows, the File Explorer is typically used to connect to shares on remote computers. On MacOS there is generally a *Connect to Server...* option inside one of the Finder menus. For more technical details on connecting to the shared network folders, you may need to consult the documentation for the operating system running on the remote computer.

On most operating systems, the connection is established by supplying the credentials along with the name of the shared resource which can usually be specified by using the IP address and target share name. The table below lists the shares created by the MotaVision system. When connecting from a remote computer in your own environment, you'll use the IP address assigned to the MotaVision system instead of the 192.168.123.1 used in this example.

\\192.168.123.1\Video	Network share of the directory where video files are stored
\\192.168.123.1\Image	Network share of the directory where overlay image files are stored
\\192.168.123.1\Script	Network share of the directory where system script files are stored

Export Content

Opens the media export dialog for copying files from the MotaVision system to external storage devices. When opened, a list of all media files on the MotaVision system is displayed. Each has a checkbox for selecting which files should be exported. The export destination can be set to a USB drive, network drive, or any other storage location recognized by the operating system.

Import Drive

When a removable drive is connected to the system, a corresponding **Import Drive** entry will appear on this menu, allowing media files on the storage device to be viewed, selected, and imported into the MotaVision system.



View Help

Opens the integrated help viewer. The standard F1 help key is also available from anywhere in the application. When F1 is pressed, the help viewer will open to the most relevant topic depending on the current activity.

Check for Updates (*Setup Mode Only*)

Used to check for software updates. When an update is available, the version information and release notes will be displayed along with action buttons that can be used to either install the update or postpone the update until the next manual check.

NOTE: MotaVision does NOT check for or apply updates automatically to ensure the update dialog will never be displayed unexpectedly.

About

Opens the About dialog which displays the software version and software license details along with other general information.



Video Control

The controls on this tab are used for testing and verifying different video playback configurations. Each command and option available here can be specified via script or network commands to trigger video playback with the same parameters and effects.

You can think of this tab as a video playback sandbox where you can experiment with all of the different controls and options to determine the ideal parameters to use when triggering video playback from a script or via network commands.

Player Status

Now Playing	Displays the filename of the active playback video
Video Length	Displays the total length of the active playback video
Player Clock	Displays the current time of the active playback video
Sync Group	Displays the sync group if the screen is included in a synchronized playback group
Playing Next	Displays the filename of the next video queued to being playback when the current video playback is finished
Video Looping	Indicates whether the active playback video was started with the loop flag

Playback Control

Stop	Click to stop video playback on the selected screen
Pause/Resume	Click to pause or resume video playback on the selected screen
Restart	Click to restart playback of the current video on the selected screen
<< (skip backward)	Click to skip backward by the configured number of seconds in the current video on the selected screen
Skip Seconds	Set the number of seconds to skip when using the skip backward and skip forward controls

>> (skip forward)	Click to skip forward by the configured number of seconds in the current video on the selected screen
-------------------	---

Position Control

Seek	Click to seek the current video playback to the configured Hours/Minutes/Seconds/Frames location
Capture	Click to copy the current video playback position to the Hours/Minutes/Seconds/Frames values
Zero	Click to set all seek position Hours/Minutes/Seconds/Frames values to zero
Hours	Set the seek position hours
Minutes	Set the seek position minutes
Seconds	Set the seek position seconds
Frames	Set the seek position frames

Screen Volume

Use the slider to adjust the volume level of the current video playback.

Screen Wipe Control

Wipe Window	Click to wipe the selected screen with the configured wipe color
Wipe Color	Click to configure the screen wipe color
#XXXXXX	Displays the #RRGGBB value of the configured wipe color

Video Files

Displays the file name, size, and timestamp of all video files that have been loaded onto the system.

File Control

Refresh Video Files	Rescan system storage and refresh the Video Files list
---------------------	--

Play Selected File	Begins playback of the selected video file
Fade Selected File	Begins a fade transition from the currently playing video file to the selected video file
Queue Selected File	Queue the selected video file for playback when the currently playing video is finished
Delete Selected File	Deletes the selected video file from system storage
Loop Playback	Enable to loop playback for video files triggered via the File Control buttons

Start Time and End Time

These numeric fields can be used to set the starting and ending time for playing a portion of video file.

Fade Control

The **Step Size** and **Step Delay** controls are used to set the cross fade step size and delay between steps when fading between two videos. There are 256 alpha steps available when fading between videos. The step size controls how much the alpha blend increases for each step. The step delay controls how long each alpha step persists before the transitioning to the next step. Different values can be combined to generate faster or slower fade transitions.

The table below provides example values for a slow transition and a fast transition. You can experiment and test with different values to find the visual effect that looks best for your particular situation.

Speed	Step Size	Step Delay	Alpha Fade Transition Time
Slow	1	10	2560 msec (2.56 seconds)
Fast	5	10	510 msec (0.51 seconds)

The fade duration can be calculate with the following formula:

$$(256 \div \text{step_size}) \times \text{step_delay} = \text{total milliseconds}$$

Sync Group

The Sync Group mechanism provides a way to synchronize playback across two or more video screens when video loop playback will be used. This is especially useful for something like video for digital eyeballs. All screens in the group will play their designated video, and when video playback on all group screens has completed, all of the videos will be restarted again at the same time. This playback restart synchronization will ensure that the videos are synchronized at the beginning of each playback loop. This

will prevent very low level timing fluctuations from accumulating which would otherwise cause the looping videos to slowly drift further and further out of sync.

Play Volume

This slider sets the initial volume that will be used when video playback is started.



Overlay Image

The controls on this tab are used for applying overlay images on top of the video playback window. Image files with alpha transparency can be used to allow the underlying video to show through in specific locations for presenting framed video or other creative display effects. Solid color images can be created and applied on the fly with full control of the alpha component. Solid colors can be applied or removed instantly, or applied gradually with the fade out and fade in controls.

You can think of this tab as an overlay image sandbox where you can experiment with all of the different controls and options to determine the ideal parameters to use when applying overlays from a script or via network commands.

Image Controls

Refresh Image Files	Rescan system storage and refresh the Image Files list
Apply Selected Image	Loads the selected image into the overlay image window
Delete Selected Image	Deletes the selected image file from system storage
Clear Overlay Image	Removes any image or solid color overlay that is currently applied and makes the overlay image window fully transparent

Solid Color

Select Solid Color	Click to configure the color used when creating the solid color overlay
#XXXXXX	Displays the #RRGGBB value of the configured solid color
Alpha Channel	Set the value of the alpha channel used when creating the solid color overlay
Apply Solid Color	Creates and applies an overlay using the configured solid color
Clear Solid Color	Removes any image or solid color overlay that is currently applied and makes the overlay image window fully transparent

Fade Out Solid Color	Begins an overlay transition from fully transparent to the configured solid color
Fade In Solid Color	Begins an overlay transition from the configured solid color to fully transparent

The **Fade Step Size** and **Fade Step Delay** controls are used to set the fade step size and delay between steps when fading in and out solid color overlays. There are 256 alpha steps available when fading in and out solid colors. The step size controls how much the alpha blend increases for each step. The step delay controls how long each alpha step persists before the transitioning to the next step. Different values can be combined to generate faster or slower fade transitions.

The table below provides example values for a slow transition and a fast transition. You can experiment and test with different values to find the visual effect that looks best for your particular situation.

Speed	Fade Step Size	Fade Step Delay	Alpha Fade Transition Time
Slow	1	10	2560 msec (2.56 seconds)
Fast	5	10	510 msec (0.51 seconds)

Image Files

Displays the file name, size, and timestamp of all image files that have been loaded onto the system.



Overlay Text

The controls on this tab are used for applying overlay text on top of the video playback window.

You can think of this tab as an overlay text sandbox where you can experiment with all of the different text font, size, color, and style options to determine the ideal parameters to use when applying overlay text from a script or via network commands.

Text Message

Field for entering the text message used to generate the text overlay.

Overlay Control

Apply Text	Creates the text overlay using the configured message, font, scale, alpha, alignment, and rectangle values
Clear Text	Clears the text overlay and restores full transparency

Font Selection

Select Font	Click to select the text font used to draw the text message on the overlay
Select Color	Click to configure the text color used to draw the text message on the overlay
#XXXXXX	Displays the #RRGGBB value of the text color used to draw the overlay text

Text Scale

When working with very high resolution displays such as 4K (3840 x 2160), standard font sizes may still appear small and difficult to see. The text scale can be set to values between 1.0 and 10.0 to significantly increase the text size when required.

Alpha

The alpha channel can be set to any value between 0 (fully transparent) and 255 (fully opaque) to control how the text appears when overlaid on top of the video.

Text Alignment

Select **Left** to left align the overlay text, **Center** to center the overlay text, or **Right** to right align the overlay text.

Text Rectangle

This field can be used to define an optional rectangle to control exact placement of the text message on the screen. The format is left,top,width,height in screen pixels. The default value 0,0,0,0 will create a text rectangle the same size as the corresponding video window.

It is also possible to provide only the left and top values, leaving the width and height values empty or zero. This will create the rectangle at the left,top values provided while using the bottom right corner of the screen for the width and height values.

Script/Network Command

Each time the **Apply Text** button is clicked, this read-only field will be updated with the exact script/network command used to generate the text message programatically.

Font Preview

This field displays sample text using the currently configured font selection.



This tab is where everything comes together. Video files, overlay images, and text content are combined with a simple yet comprehensive scripting language to create scripts that power seamless and automatic system control.

See the **Command Reference** help page for a complete list of all script commands available for controlling the system.

See the **Example Scripts** help page for sample scripts that show how to use the script language for complex system control.

Editing Area

The left section contains a specialized text editor for creating and editing scripts. The editor displays line numbers in the left margin for quick location reference in long or complex scripts. The line number will also displayed in popup messages if syntax or other errors are detected when testing a script.

The editor also features syntax highlighting that uses different colors for the different types of commands:

- **Red** - Used when the first word on a line is not a valid command or keyword
- **Blue** - Used for commands related to script control
- **Purple** - Used for screen commands that are valid in scripts or network messages
- **Cyan** - Used for system level commands
- **Yellow** - Used for script variable and pool declarations
- **White** - Used for script jump labels
- **Gray** - Used for comment lines that start with #

Save Script

Click this button to save the contents of the script editor.

Test Script

Click this button to test the script in the editor. This is especially useful when creating or editing scripts. During execution, the test logic will check for syntax or other errors, displaying a popup a message that includes the line number and error details if problems are detected.

Script Name

This field holds the name of the script. Creating new copies or different versions of a script can be achieved simply by changing the name and saving the script.

Editor Controls

Refresh Script List	Rescan system storage and refresh the list of script files
Edit Selected Script	Loads the selected script in the editor area
Run Selected Script	Loads and runs the selected script on the system script engine
Rename Selected Script	Rename the selected script
Delete Selected File	Deletes the selected script from system storage
Double Click	Select the operation to perform when a file in the script list is double clicked

File Name Helper Buttons

When editing scripts, manually typing the exact name of video, image, or script files in script commands can be tedious. The file name insertion buttons allow you to view the available files of each type that are available on the system, and select and insert the name automatically. The name will be inserted at the current cursor position.

Insert Video File Name	Select an available video file and insert the name in the editor
Insert Image File Name	Select an available image file and insert the name in the editor
Insert Script File Name	Select an available script file and insert the name in the editor

Set Editor Font

Click this button to select the text font used in the script editor.

Create New Script

Click this button to begin working on a new script. The contents of the script editor and the Script Name fields will both be cleared.

Script Files

The right area displays the name, size, and timestamp of all script files available on the system.



This tab provides the script engine status display and script scheduling controls used to configure the system for fully automated and interactive video playback.

Active Scripts

This table shows the name of any scripts that are currently running. In addition to the system script engine, each screen in the system has a dedicated script engine that runs independently. This allows for complex configurations where each video screen operates with its own dedicated control script.

For example, a bootstrap script could be created and assigned in the AutoRun configuration to run at system startup. This script would launch one script on the Screen 1 script engine and a different script on the Screen 2 script engine.

The script running on Screen 1 could be a simple looping script that plays through a sequence of video files and repeats.

The script running on Screen 2 could be interactive, waiting for input events, playing video when triggered, and then waiting for the next input event.

The example scenario effectively creates two independent video playback engines. One playing videos in a loop on one screen while videos triggered by input events play on another screen.

Event Triggers

The Event Triggers table provides a mechanism to assign script files to execute when specific events occur. When creating or modifying an event, the following event types are supported:

- **NETWORK** - Responds to the network TRIGGER command
- **JOYSTICK** - Responds to joystick button press events
- **INPUT** - Responds to inputs from the integrated IO controller
- **CLOCK** - Activates at a specific time of day

For all event types, the script to execute can be selected in the list of available scripts. For discrete input events, an input number selector will be displayed. For clock events, a time-of-day selector will be displayed.

To modify an existing Event Trigger, double click the row in the Event Trigger list.

Create Event Trigger	Click to create a new Event Trigger
Delete Event Trigger	Click to delete the selected Event Trigger

MIDI Triggers

The MIDI Triggers table provides a mechanism to assign script files to execute at specific MIDI timecode markers. When creating or modifying an event, the script to execute is selected in the list of available scripts and the timecode marker when the event should be triggered is selected with the hour, minute, second, and frame controls.

To modify an existing MIDI Trigger, double click the row in the MIDI Trigger list.

Create MIDI Trigger	Click to create a new MIDI Trigger
Delet MIDI Trigger	Click to delete the selected MIDI Trigger



The controls on this tab are used to view and manage the system screen configuration.

Multiple Displays on Windows

Windows treats all connected displays as one large virtual desktop space. The primary display is positioned at coordinates 0,0 which are the X and Y values that correspond to the left and top values. Additional displays are arranged around the primary display at positive or negative X and Y offsets depending on their arrangement in the display settings.

For example, a monitor placed to the right of the primary display will have a positive X offset while one placed to the left will have a negative X offset. A monitor placed below the primary display will have a positive Y offset, while one placed above it will have a negative Y offset.

Selection of the primary monitor and the arrangement of additional monitors can be managed in the Windows settings. For quick access, right click the empty space on the Windows desktop and select **Display Settings** from the popup menu. The display settings can also be accessed directly in the Windows Settings screen. Select **System** from the main list of categories and then select **Display** on the left side menu.

Under the Windows virtual desktop model, MotaVision does not play video on specific monitors. Instead, logical screens are created with display windows that correspond to the exact virtual location and size of each individual monitor included in the virtual desktop, and video is played on those screens. With an understanding of the virtual desktop and display layout mechanism, it is possible to create video screens of any size and at any location on the Windows virtual desktop.

MotaVision Automatic Detection

By default, MotaVision will detect all displays connected to the system and create a video screen for each one using the full pixel width and height reported by the system. The auto-detection is normally static and consistent, meaning the same screen setup will be created each time the application is launched.

Display detection and assignment on Windows is usually seamless, but it can also sometimes be chaotic. For the optimal configuration and control of the MotaVision system, it will be helpful to understand how both MotaVision and Windows handle detecting and configuring connected displays.

Manual Screen Configuration

Changes to the hardware configuration or random low level detection timing logic can affect the logical creation and ordering of the screens. If this happens, Screen 1 and Screen 2 could become swapped, with content for one playing on the other, and vice versa. Manually defining the location and size of each screen can be helpful if auto-detection is not consistent.

It is also possible, and sometimes highly useful, to create screens that are smaller than the size of a full display screen. For example, in the MotaVision test and development environment, we create four quarter size HD video screens on a secondary monitor that is located to the left of the primary monitor:

Screen	Name	Left	Top	Width	Height
1	Upper Left	-1920	0	960	540
2	Upper Right	-960	0	960	540
3	Lower Left	-1920	540	960	540
4	Lower Right	-960	540	960	540

This configuration simplifies development and testing by allowing the MotaVision user interface to appear on the primary monitor, with four small video screens on the secondary monitor. This allows testing the features and functional performance of a five monitor configuration without requiring five physical monitors to be connected.

Manual configuration could also be used to support other custom setups. For example, with the right very high-end hardware, you could attach four 4K displays arranged in a 2x2 square, creating a virtual 8K desktop that is 7680 x 4320. You could then create a single 8K screen in MotaVision and play 8K videos seamlessly across the combined multi-monitor screen.

Create Video Screen	Click to create a new Video Screen
Delete Video Screen	Click to delete the selected Video Screen

To modify an existing Video Screen, double click the row in the Video Screen list.



This page provides details for all commands available for controlling the MotaVision platform. Commands can be sent over the network using UDP messages, or combined with labels, variables, and control statements in scripts to provide fully automatic control.

Script Labels, Variables, and Control Statements

These commands are only valid in script files. They are used to define labels, variables and statements that can be used to control script execution and flow.

All commands, label names, variable names, and other names are case-insensitive, so jump MyLabel, JUMP MYLABEL, JUMP mylabel, etc. are all the same.

Command	Description
# comment	Lines that start with the # character are comment lines and are ignored by the script engine.
:mylabel	Defines a script label named mylabel which can be used as the target of jump and compare commands.
jump mylabel	Jump to the script label named mylabel and continue script execution from that point.
var counter 15	Declare a numeric variable named counter with the value of 15. If the initial value is omitted, the default 0 will be used.
set counter 0	Set the variable named counter to the value 0.
inc counter	Increment the variable named counter by 1.
dec counter	Decrement the variable named counter by 1.
pool mypool 1 10	Declare a number pool named mypool with values from 1 to 10. The values in the pool are created and will be returned in numerical order by default.
fetch mypool counter	Fetch the next number from the pool named mypool and store in the variable named counter.
shuffle mypool	Shuffle the numbers in the pool named mypool so subsequent fetch operations get a random value from the pool while ensuring the same number is never returned twice in a row.
random counter 1 4	Generate a random number between 1 and 4 and store the value in the variable named counter.

compare counter 5 jump mylabel	Compare the variable named counter with 5 and jump conditionally to the label mylabel. The jump condition can be one of: • JUMPEQ = jump if counter is equal to 5 • JUMPLT = jump if counter is less than 5 • JUMPGT = jump if counter is greater than 5 • JUMPLTE = jump if counter is less than or equal to 5 • JUMPGTE = jump if counter is greater than or equal to 5
recycle	Restart the current script from the beginning.
sleep 5000	Suspend execution of the current script for 5000 msec (five seconds)
wait event index	Suspend execution of the current script until the specified event and index are received: • wait VIDEO 1 = wait until the video on screen 1 has finished playback • wait QUEUE 2 = wait until the queued video on screen 2 has started playback • wait FADER 3 = wait until the fade cycle on screen 3 has finished • wait NETWORK 4 = wait until the network message "trigger 4" is received • wait STICK 5 = wait until joystick button 5 is pressed • wait INPUT 6 = wait until the hardware input 6 is detected (not yet implemented pending choice of hardware to support)
exit	End processing of the current script.

System Control Commands

These commands are valid in script files and network messages. They control system components not specific to any screen.

Command	Description
script "Script Name" [screen #]	Load and begin execution of the specified script. The screen number is optional. When the screen number is omitted, the script will run on the system script engine. When a screen number is included, the script will run on the script engine dedicated to the specified screen. This mechanism allows each screen to operate as a completely independent video player, as each screen can have a different script active at any time. When a script runs on the dedicated script engine of a specific screen, the screen= parameter is ignored and can be omitted in all of the Screen Control Commands, and the screen where the script is running will automatically be used. Also, the @

	symbol can be used in place of a screen number in the VIDEO, QUEUE, and FADER wait commands. This allows the script to wait for events associated with the target screen, allowing the same script to be used unchanged on any screen.
trigger index	Send a network event trigger with the specified index value.
showcon	Show the control console window.
hidecon	Hide the control console window.
terminate	End execution of any script that may be running.
transmit address port "message"	Transmit a UDP network message to the specified IP address and port.

Screen Control Commands

These commands are valid in script files and network messages. They control the video and overlay windows of an individual screen.

When used in a script, the screen= parameter MUST be included.

When received from the network with no screen= parameter, the commands will be applied to the screen assigned to the network port where the command was received. Network messages can include the screen= parameter, allowing commands to be sent to any screen from either network port.

Some commands include optional arguments which are described below.

Command	Description
play "VideoFile.mp4" start= end= volume= loop= sync=	Start playback of the specified video file. • Add start=HH:MM:SS.FF to set the video start position. • Add end=HH:MM:SS.FF to set the video end position. • Add volume=[0-100] to set playback volume. • Add loop=1 to loop the video. • Add sync=[1-32] to specify the playback sync group.
queue "VideoFile.mp4" start= end= volume= loop=	Queue the specified video file for seamless playback immediately after the file currently playing. • Add start=HH:MM:SS.FF to set the video start position. • Add end=HH:MM:SS.ff to set the video end position. • Add volume=[0-100] to set playback volume. • Add loop=1 to loop the video.
fadeto "VideoFile.mp4"	Start playback and fade from the currently playing video to the specified video file.

start= end= volume= loop= sync= step= delay=	• Add start=HH:MM:SS.FF to set the video start position. • Add end=HH:MM:SS.FF to set the video end position. • Add volume=[0-100] to set playback volume. • Add loop=1 to loop the video. • Add sync=[1-32] to specify the playback sync group. • Add step=[1-255] to specify the fade step size • Add delay=[1-1000] to specify the fade step delay
seek HH:MM:SS.FF	Seek to the specified time in the currently playing video.
stop	Stop playback.
pause	Pause playback.
resume	Resume playback.
restart	Restart playback of the current video file from the beginning position.
wipe color=	Wipe the video window with the configured screen color. • Add color=#RRGGBB to wipe the video window with a specific color.
volume 75	Set the playback volume to 75 percent.
text "message" font= color= align= rect= scale=	Draw the specified message on the video overlay. Optional values will default to whatever values were last used if not specified. Use "" (empty quotes) to clear any existing text message. • Add font="[name,size]" to set the font name and size. • Add color=[#AARRGGBB] to set the font color. • Add align=[left center right] to set the text alignment. • Add rect=[x,y,w,y] to set the text location. • Add scale=[1.0 - 10.0] to scale the text size for high DPI screens.
color #AARRGGBB	Create a solid color overlay with the alpha, red, green, and blue hex values of the specified color code.
image ImageFile.png	Create a video overlay from the specified image file.
alpha	Remove any solid color or image overlay currently applied to the overlay window and revert to full transparency.
clear	Clear any text and color or image overlays and return the overlay window to full transparency.
fadeout #RRGGBB step delay	Begin a fade out from fully transparent to fully opaque with the specified #RRGGBB color, step size, and step delay (msec).

fadein #RRGGBB step delay	Begin a fade in from the fully opaque to fully transparent with the specified #RRGGBB color, step size, and step delay (msec).
status	Query the status of the video player.



Example Scripts

This page provides example scripts for playing videos, displaying text, and using labels, variables, and conditional control statements. Each example includes comments to explain what the script is doing.

Video Eyeballs

This simple script is all that is required to create a pair of synchronized video eyeballs.

```
# These first two lines are optional and simply show how to draw a text label on each screen
text "Left Eye" screen=1 color=#ffccccc align=center
text "Right Eye" screen=2 color=#ffccccc align=center

# These are the lines that actually play, loop, and synchronize the two eyeball videos
play LeftEye.mp4 screen=1 volume=0 loop=1 sync=1
play RightEye.mp4 screen=2 volume=0 loop=1 sync=1
```

Each eyeball video is played on a different screen, with the loop flag enabled on both. Each also has sync=1 which will assign them both to Sync Group one which will automatically synchronize the playback loop. When either video is finished, that player will wait for the other player in the group to finish. Once both have finished, both players will be restarted at the same time. Assuming the videos are the same length, the synchronization and restart happens in microseconds, effectively creating a perfectly synchronized endless loop on the two screens.

Fader Cycle and Wait

This more complex script starts playback of a video, and then performs a dark blue fade-out and fade-in three times before stopping video playback.

```
# Start a video playing
play MusicVideo.mp4 screen=1 volume=0

# Clear the text display and set the text color and font for subsequent text commands
text "" screen=1 color=#ffddddd font="Arial,32"

# Declare a variable named counter with an initial value of zero
VAR counter 0

# This is our jump label for the counter comparison
:REPEAT

text "Two Second Wait" screen=1
sleep 2000

text "Calling FadeOut" screen=1
```

```

fadeout #0000dd 4 20 screen=1
wait fader 1

text "Two Second Wait" screen=1
sleep 2000

text "Calling FadeIn" screen=1
fadein #0000dd 4 20 screen=1
wait fader 1

text "Fade Cycle Complete" screen=1
sleep 2000

# Increment the counter
inc counter

# Compare the counter and jump to the REPEAT label above if less than three
compare counter 3 JUMPLT REPEAT

# After three fade cycles we write the finish text and stop the video
text "Fade Test Finished" screen=1
stop screen=1

```

The script above uses **wait fader 1** several times to pause execution of the script until the fade command completes. After the fade-out and fade-in cycle, the script increments the counter variable which was declared near the top. If the counter is less than three, the script jumps to the :REPEAT label. When the counter reaches three, the conditional jump will be skipped and the script will continue to the next line where the finish messages is displayed and the player stop command is executed.

Random Attraction Video

This script creates a pool with the numbers 1-4 and shuffles the pool to generate a random order. It then fetches numbers from the pool and plays a different video for each possible value. Using a shuffled number pool is a great way to achieve random playback order while ensuring every video in the pool is played. Once all numbers in the pool have been returned, the pool will automatically be shuffled again on the next fetch, ensuring random but complete playback of all videos. After each shuffle, the pool will ensure the next value fetched is not the same as the last value fetched before the shuffle, so the same video will never play twice in a row.

```

# define a number pool named NumberPool with the values 1, 2, 3, and 4
POOL NumberPool 1 4

# shuffle the pool to create random playback order
SHUFFLE NumberPool

# define a variable named AttractVideo
VAR AttractVideo 0

:REPEAT

# fetch the next value from NumberPool and store in the AttractVideo variable
FETCH NumberPool AttractVideo

```

```

# compare AttractVideo to the four possible random values and jump to the corresponding label
COMPARE AttractVideo 1 JUMPEQ ONE
COMPARE AttractVideo 2 JUMPEQ TWO
COMPARE AttractVideo 3 JUMPEQ THREE
COMPARE AttractVideo 4 JUMPEQ FOUR

# play attract video one
:ONE
play "AttractOne.mp4" screen=1
WAIT VIDEO 1
JUMP REPEAT

# play attract video two
:TWO
play "AttractTwo.mp4" screen=1
WAIT VIDEO 1
JUMP REPEAT

# play attract video three
:THREE
play "AttractThree.mp4" screen=1
WAIT VIDEO 1
JUMP REPEAT

# play attract video four
:FOUR
play "AttractFour.mp4" screen=1
WAIT VIDEO 1
JUMP REPEAT

```

This script uses multiple labels to control playback flow. After fetching a value from the pool, the number is compared to the four possible results, and the JUMPEQ command is used to jump to one of four different labels. Each label plays a different attraction video, waits for video playback to complete, and then jumps back to the REPEAT label.

Random Playback Screen

This script generates a random number between 1 and 4 and plays a short video on one of four video screens based on the random result. Unlike a number pool that has been shuffled, the random number generation is truly random just like rolling dice.

```

# define a variable named ShowScreen
VAR ShowScreen 0

:REPEAT

# generate a random number between 1 and 4 and store in the ShowScreen variable
RANDOM ShowScreen 1 4

# compare ShowScreen to the four possible random values and jump to the corresponding label
COMPARE ShowScreen 1 JUMPEQ ONE

```

```
COMPARE ShowScreen 2 JUMPEQ TWO
COMPARE ShowScreen 3 JUMPEQ THREE
COMPARE ShowScreen 4 JUMPEQ FOUR
```

```
# play short video on screen one
:ONE
play "LeftEye.mp4" screen=1
WAIT VIDEO 1
JUMP REPEAT
```

```
# play short video on screen two
:TWO
play "LeftEye.mp4" screen=2
WAIT VIDEO 2
JUMP REPEAT
```

```
# play short video on screen three
:THREE
play "LeftEye.mp4" screen=3
WAIT VIDEO 3
JUMP REPEAT
```

```
# play short video on screen four
:FOUR
play "LeftEye.mp4" screen=4
WAIT VIDEO 4
JUMP REPEAT
```

This script uses multiple labels to control playback flow. After generating the random number, the random value is compared to the four possible results, and the JUMPEQ command is used to jump to one of four different labels. Each label plays a video on a different screen, waits for video playback to complete, and then jumps back to the REPEAT label.

Seamless Loop

This script demonstrates using the QUEUE command and the WAIT QUEUE event to seamlessly loop two videos.

```
# Example of looping two videos seamlessly using the WAIT QUEUE event

# start by playing the first video immediately
play TestLeftEye.mp4 screen=1

# queue the second video to play immediately after the first
queue TestRightEye.mp4 screen=1

:REPEAT
# wait for the queued video to start
wait queue 1

# queue the first video
queue TestLeftEye.mp4 screen=1
```

```
# wait for the queued video to start  
wait queue 1  
  
# queue the second video  
queue TestRightEye.mp4 screen=1  
  
# jump to the repeat label  
jump REPEAT
```

The key feature here is the WAIT QUEUE event which pauses the script until the queued video begins playing. At that point the next video can be queued, providing a way to seamlessly loop any number of videos.