

# Video Kiosk – Android User's Manual

Burningthumb Studios

---

# Table of Contents

<b>Introduction</b> .....	<b>1</b>
System Requirements .....	1
Free Trial Mode .....	2
Purchasing and Licensing.....	2
<b>Installing and Configuring Video Kiosk</b> .....	<b>7</b>
Installation .....	7
About permissions .....	9
Video Kiosk Settings .....	10
Using Settings XML File.....	12
<b>Features Overview</b> .....	<b>13</b>
Video Kiosk's default behavior .....	13
Video Kiosk Features.....	14
<b>Understanding Screen Display Modes</b> .....	<b>18</b>
Full Screen Mode.....	18
Split Screen Mode .....	18
Touch Screen Interactive Kiosk Display Mode.....	19
<b>Working with Media Types</b> .....	<b>20</b>
Supported Media Formats .....	20
Displaying Video .....	20
Displaying Images.....	20
Using URL Shortcuts to display Content.....	20
<b>Cloud Download</b> .....	<b>24</b>
How to Configure and Use Cloud Download Update .....	24
<b>Controlling Playback</b> .....	<b>27</b>
Default Playback Behavior.....	27
Controlling Playback Order .....	27
Controlling Playback Time using a Schedule .....	31
Controlling Playback using device location.....	40
Controlling Playback using Motion Detection.....	44
Monitoring Battery State .....	44
Changing the Playback Mode.....	44
<b>Controlling the Display</b> .....	<b>46</b>
View Options - Screen Orientation, Timers and Transitions .....	46
Overlays .....	48
Background Images and Audio .....	50
Controlling Display using file name meta data .....	51
Legacy Support Display Features .....	54
<b>How to Convert to Split Screen Display Mode</b> .....	<b>55</b>
<b>How to Convert to Touch Screen Interactive Kiosk Display Mode</b> .....	<b>64</b>
<b>Controlling the Device</b> .....	<b>69</b>

TV Control using HDMI CEC (Android v5 and later) .....	69
Keyboard and Remote Control Features .....	71
Touch Gesture Control.....	71
Gamepad Control Features .....	72
<b>Reliability Features .....</b>	<b>73</b>
Handling changes to the playback loop.....	73
Handling unplayable content.....	73
Behavior on wake from sleep.....	73
Error Detection Automatic Restart.....	74
Periodic Automatic Restart .....	74
<b>Access Management .....</b>	<b>75</b>
Access Management recommendations.....	75
<b>Remote Management .....</b>	<b>80</b>
How to enable device status reporting.....	80
HTTP vs HTTPS considerations.....	80
Status Report Post Content: Supported POST Key Value Pairs Sent .....	81
Responses: Supported POST Key Value Pairs in Response .....	83
Example of Use.....	85
<b>Configuring the Settings .....</b>	<b>87</b>
<b>Troubleshooting.....</b>	<b>103</b>
<b>Appendix A – Making Video Kiosk a System App .....</b>	<b>104</b>
<b>Appendix B – Making Video Kiosk the device owner .....</b>	<b>105</b>
<b>Appendix C – Video Kiosk Settings XML File .....</b>	<b>107</b>
<b>Appendix D – Built-in Widget Settings XML File.....</b>	<b>114</b>
<b>Appendix E – Registering uncertified devices .....</b>	<b>117</b>
<b>Appendix F – Basic How Tos .....</b>	<b>118</b>
Setting the Android device Time Zone .....	118
Copying a folder from your computer to an Android device .....	118
Finding the Google account for your Android Device.....	119
<b>Appendix G – WebDAV .....</b>	<b>120</b>
<b>Appendix H – Personal Cloud .....</b>	<b>121</b>
The script on Github.....	121

# Introduction

Turn your Android device into an easy to use, reliable, robust, secure Android kiosk to loop video, images, and web pages in minutes. Easy to use, you'll have your Android TV | tablet | phone playing videos, images, and web pages and running unattended, reliably and robustly with minimal set up. It comes with optional advanced features such as remote Cloud Download, playback scheduling, secure Kiosk Mode for access management of interactive displays, remote management, overlays, backgrounds and flexible screen displays – full screen or split screen as well as touch screen interactive kiosk capability.

This manual will cover both basic installation as well as how to use Video Kiosk's more advanced features.

## System Requirements

**System Requirements:** Android 4.x or later

Some advanced features, notably HDMI-CEC, require Android v5+ or Android v6+. Please refer to the HDMI-CEC section of this document for more information on the system requirements for those features.

**Compatibility:** Android phones, tablets, TV Sticks

Video Kiosk has been tested and deployed on the following devices:

- TV Sticks (tested on CX-919, Fire TV, i68, MK808, MK809, MK812, MK903, i68, nVidia Shield TV, Razer Forge TV, Pine A64, and many more)
- Tablets (tested with Acer, rooted Kindle Fire, generic sub \$100 tablet, and many more)
- Phones (tested with Android v5, v6, v7, v8, v9 phones)

**Note:** To avoid the black flash problem found on Amlogic SoC, enable Textured Video Views and / or Pause playback in the Video Kiosk settings. Alternatively, use a different TV Box, such as a Rockchip, Alwinner, or nVidia, based TV box (like i68). Contact us if you are having an issue with your TV box.

**Internet Access.** Video Kiosk can run fully self-contained and internet access is not required for most features. An Internet connection is only required:

- to get the app from the Playstore
- to activate (and deactivate) licenses
- to update your Content Loop over the cloud, if you're using this feature
- to schedule Playback using a Google Calendar, if you're using this feature

Once the license is active, you may turn off internet access on the device.

## Free Trial Mode

When you download Video Kiosk, it is in trial mode. All features are fully functional, but a watermark will appear over your content. To remove this, purchase a license.

## Purchasing and Licensing

**Video Kiosk is licensed on a per device basis, meaning a license must be purchased and activated for each device on which Video Kiosk is used.**

There are three ways to purchase a license:

- **In-App purchase** through the Google Play Store on the Android device
- Purchase through our **Web Store** on any computer or Android device
- By **invoice** for large volume purchase of 25 or more licenses

### Purchase Option 1: In-App purchase through the Google Play Store

To purchase in the app,

1. If you haven't done so already, **download** the Video Kiosk app from the Google Play Store on the device.
2. Open Video Kiosk.
3. To **purchase** and **activate** a license for a device, click the **Purchase Button** on the **License Activity** screen and follow the In-App purchase steps. The license will automatically be activated when the In-App purchase completes.

### Purchase Option 2: Purchase through our Web Store

Purchasing and licensing through our Web Store is an easy 4-step process:

1. Download the Video Kiosk app on to the device.
2. Decide which license management option you will be using.
3. Purchase Licenses through our Web Store.
4. Activate the Licenses.

#### Step 1: Download Video Kiosk

**Download** the Video Kiosk app on to the device.

You can do this:

- Through the Google Play Store on the Android Device, or
- Sideload it by downloading the .apk from:  
<https://burningthumb.com/download/videokiosk/latest/VideoKiosk-videokiosk-release.apk.zip>

## Step 2: Decide on a Volume purchase License Management Option

You can manage your licenses using either of these two options:

- **Email ID.** Add an account (Email ID) to each of your devices or
- **License File.** Place a license file, that we send to you, on your devices at the path  
/sdcard/videokiosk.license

Regardless of which option you choose, you can add more licenses to your account by purchasing more from our site.

### *Email ID License Management Option*

Before deciding to use the Email ID License Management Option, here are some things to consider:

- When you use the Email ID License Management Option, you need to add an Account with that Email address to each Android device.
- Some Android do not support adding accounts, and which case, the **Add Account** button from the Video Kiosk license activity screen will be hidden. If this is the case with your devices, use the **License File License Management Option** instead. To get the app, sideload Video Kiosk from our site.
- Before you decide to use the **Email ID License Management Option**, make sure you can add the account you plan to use to the device *before* purchasing licenses for the account.

You can add an account to an Android device in a number of ways. (See Android Basics at <https://burningthumb.com/android-basics/>) The **easiest way** to add them is to:

- a. Install and open Video Kiosk,
- b. Grant the requested permissions,
- c. Add the requested account.

### *License File License Management Option*

Before deciding to use the Email ID License Management Option, here are some things to consider:

- When you use the License File License Management Option, you need to copy the License File to each Android device.

## Step 3: Purchase Licenses through our Web Store

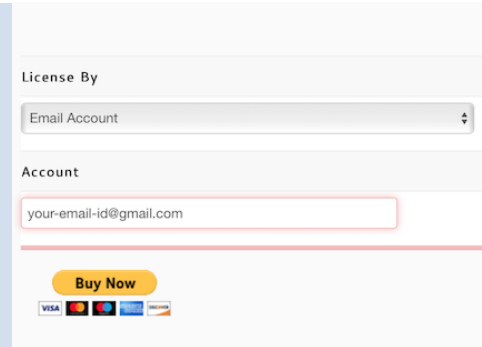
**Before purchasing, determine which license management option you will be using.**

To purchase volume licenses, do the following:

1. Go to the Volume Purchase page by clicking the Volume Licensing button on the Video Kiosk Page. (This can be on any computer. It doesn't have to be on the Android device on which you will activate the license.)
2. Enter your license management option.

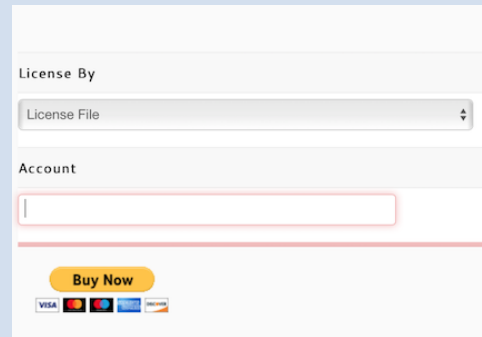
If you are using the **Email ID License Management** option,

- choose “**Email Account**” from the “**License By**” pop-up menu and
- enter the Email ID you are using into the “**Account**” field.



If you are using the **License File License Management** option,

- choose “**License File**” from the “**License By**” pop-up menu and
- leave the “**Account**” field blank.



3. Click the “**Buy Now**” button to continue.
4. On the Purchase Screen, enter the number of licenses you’re buying and click the “**Continue**” button to proceed to the Check Out.
5. Once you have checked out, you will be sent an email with the steps to activate the licenses with the License File attached, if applicable, and the licenses will be added to the license server so you can activate them.

## Step 4: Activate Volume Licenses

**You can buy the Licenses from our Web Store on any device (the Android device, your computer) but you must Activate the License on the device you are going to use it on.**

License Activation is done using the License Activity screen on your device. Due to your device’s screen resolution and orientation, it may be that the entire License Activity screen is not visible. If this happens, use touch and swipe up / down to scroll the screen contents.

To activate a license on a device, do the following:

1. If you haven't already done so, install Video Kiosk from the Play Store on to the Android device (or side load the .apk file).
2. If you’re using the **License File License Management Option**, copy the license file onto the Android device at the path /sdcard/videokiosk.license so that Video Kiosk can automatically find it.  
If you’re using the **Email ID License Management Option**, add the Email Account to the Android device.
3. Ensure that the Android device has Internet access.

4. Open Video Kiosk on the Android device. Grant any permissions requested.
5. (Optional) If you want the license to automatically expire after a preset number of days have elapsed, fill in the number of days in the “**Active Days**”.
6. On the “License Activity” screen, click the “**Activate**” button.

**Note:** The account (Google, Facebook, Email, etc.) OR videokiosk.license file and an Internet connection are only required to activate (and deactivate) licenses. Once a license is active, you may remove the account OR videokiosk.license file from the device. You may also remove the device from the Internet.

## Purchase Option 3: Volume Purchase by Invoice

To purchase more than 25 licenses and receive a volume discount, follow these instructions:

### *Decide on a license management option*

1. Decide which license management option you will be using. To **Decide on a Volume purchase License Management Option**, follow the instructions in Purchase Option 2, above.

### *Purchase Licenses*

To purchase more than 25 licenses and receive a volume discount, do the following:

1. Send an email to [burningthumb@gmail.com](mailto:burningthumb@gmail.com) that contains the following information:
  - a. The number of licenses required
  - b. If you plan to use an account on the device to manage licenses, provide the account (Google, Facebook, Email, etc.) that will be used to activate the licenses,
  - c. If you do not plan to use an account on the device to manage licenses, indicate that you require a license file (used to activate the licenses)
  - d. The email address that should receive the invoice
2. An invoice will be sent from paypal.com to the email address you provide. You can pay the invoice using any major credit card
3. Once the invoice is paid, you will be sent an email with the steps to activate the licenses with the License File attached, if applicable, and the licenses will be added to the license server so you can activate them.

### *Activate Licenses*

1. To **Activate Licenses**, follow the instructions in Purchase Option 2, above.

## Managing Volume Licenses

Licenses purchased under the volume purchase plan are floating, allowing you to move them from device to device. When a license expires or is deactivated, the license returns to the pool of available licenses and may be used to license Video Kiosk on another device. (Video Kiosk returns to Free Trial mode on the deactivated device.)

License Deactivation is done using the Video Kiosk Settings. To deactivate the license on a device, go to the Settings > License and Choose “Deactivate”.

Once the license is deactivated, you can use it on a different device.

### **Internet Access**

Internet Access as well as the License File or Email account (Google, Facebook, Email, etc.) you used to purchase Video Kiosk (through the Google Play Store or through our website) are only required to activate (and deactivate) licenses. Once a license is active, you may remove the Email account, the videokiosk.license file and internet access from the device.

### **Using uncertified devices**

For more information, refer to the Troubleshooting section.

# Installing and Configuring Video Kiosk

## Installation

Video Kiosk is designed to be a reliable, video kiosk playing media content in a loop with minimal configuration. To use Video Kiosk, follow these steps:

1. On your computer, create a Video Kiosk folder layout to contain your Media Files and any required Configuration Files and put your files in the appropriate folder on your computer. (See below for details)
2. Move your files onto your Android Device
  - using the Video Kiosk Remote Update Feature, or
  - using the Android File Transfer
3. Launch Video Kiosk.

Video Kiosk will play the Content Loop in the Media Folder, unattended. You can play videos, images and URL shortcuts, or any combination of the three.

For more information about supported media formats, refer to [Supported Media , p. ].

## How to create a Video Kiosk Folder Layout

There are three ways to create the Video Kiosk Folder Layout.

### Option 1 – Use the Sample Pack

We've created a **Media Sample pack** for you which contains the **Standard Folder Layout** and contains some sample media files. All you need to do is move it to the device. Once you have the sample playing, replace the sample media files with your media files.

1. Download the Video Kiosk Media Sample Pack from:  
<https://burningthumb.com/wp-content/uploads/video-kiosk-media-sample-pack.zip>
2. Unzip it. Inside, you'll find a "videokiosk" folder with the Standard Folder Layout.
3. Move the videokiosk folder layout onto your Android Device
  - using the Video Kiosk Remote Update Feature, or
  - using the Android File Transfer (See Appendix F – Basic How Tos)

### Option 2 – Create the Folder Layout on your computer

Create and manage your Folder Layout and media on your computer. Then move it to your Android device.

1. On your computer, create one of the three layouts. Move your media files to the appropriate folder.

2. Move the videokiosk folder layout onto your Android Device
  - using the Video Kiosk Remote Update Feature, or
  - using the Android File Transfer (See Appendix F – Basic How Tos)

### Option 3 – Create the Folder Layout on your device

You can create a folder on your device using the Android File Manager (File Browser | File Explorer) on your device or using Video Kiosk. For instructions on how to use your Android File Manager, refer to the documentation for your device.

To create a folder using Video Kiosk, follow these instructions:

1. Display the Video Kiosk Button Bar. To display the Button Bar, tap the screen, swipe down, swipe up, or press the down arrow on a controller or remote control.
2. Pick the Folder Icon. This will display the Media Folder.
3. Navigate to the location where you want to create the new folder and click the "Create Folder" button.
4. Enter the folder name and click "Confirm".

## About Video Kiosk Folder Layouts

There are **three ways to organize the files** (media files and configuration files) you create when using Video Kiosk, the Standard Folder Layout, the One-Folder Layout and the Custom Folder Layout.

### Option 1: Standard Folder Layout

When you are using all the features of Video Kiosk, you will have a number of Media files and Configuration files to manage, which may be cumbersome if they're all in one folder. For this reason, we recommend you use the Standard Folder Layout below, putting the different file types in the different sub-folders. By adopting this structure from the very beginning of your project, you will find it easier to manage your Video Kiosk files as you add more features to your projects. In addition, if you use the Standard folder layout, there is no need to configure the settings with configuration file locations since Video Kiosk will automatically find files in these locations.

The Standard Folder Layout is:

```
/sdcard
  /videokiosk
    /background
    /location
    /media
    /overlay
    /playlist
    /schedule
    /settings
    /widget
```

You'll put your media files in `/sdcard/videokiosk/media`

## Option 2: One Folder Layout

For basic use, you can create a single Video Kiosk folder and put all your Media files and all your Configuration files in it.

The One Folder Layout is:

```
/sdcard
  /videokiosk
```

You'll put your media files in `/sdcard/videokiosk`

## Option 3: Custom Folder layout

You can create your own file layout for the Configuration Files. If you use Configuration folders for your configuration files (Schedule, overlay, etc), you'll need to Configure the location of each configuration folder since you're not putting them in the default location.

To configure the Media Folder location, refer to the Video Kiosk Settings Reference Guide.

The Legacy Folder Layouts Video Kiosk also recognizes are:

```
/sdcard/<android id>
/sdcard/Movies/<android id>
/sdcard/Video Kiosk
/sdcard/Movies/Video Kiosk
```

You'll put your media files in it.

## About permissions

When you install Video Kiosk, you have always been prompted to grant the permissions that Video Kiosk requires. Beginning with Android v6, you will also be prompted at run time to grant some, but not all, of those same permissions again. If you choose to deny permissions at run time, some of Video Kiosks features will be disabled.

We recommend that you grant all of the permissions so that you don't run into problems with features you have inadvertently disabled by denying the run time permission request.

# Video Kiosk Settings

Video Kiosk offers many additional features which you can use to enhance the reliability, usability and management of your Video Kiosk, which are documented in this manual. In addition, detailed information about each setting can be found in the “Configuring the Settings” section of this manual.

## Where to Find the Settings

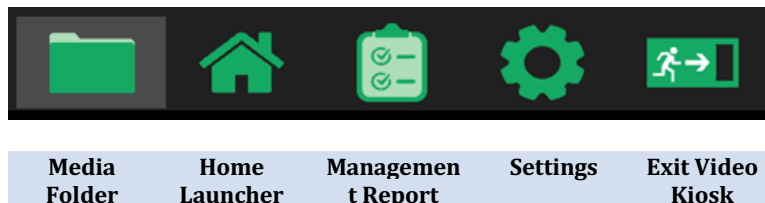
Video Kiosk’s settings are split into two areas: The Button Bar and the Settings Screen (and sub-screens). The features of each are described below.

### Button Bar

By default, the Button Bar is displayed for a few seconds when you launch Video Kiosk and then it automatically hides.

To display it again you can

- Single-tap the screen, or
- Swipe up on the screen
- Swipe down on the screen
- Press the down arrow on a controller or remote control.



The button bar allows you to:

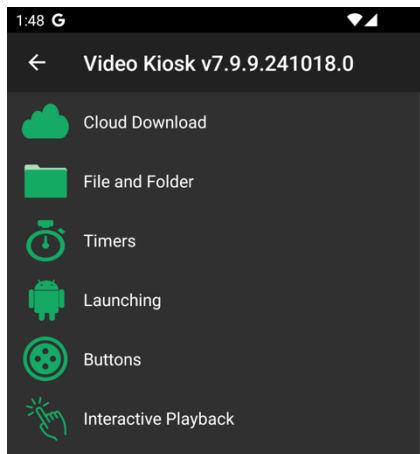
- Check or Choose the **Media folder** containing the Media Files
- Set Video Kiosk as the **Home Launcher**
- Send a **Management report**
- Open the **Settings** screen
- Open the **License Activation** screen (Only appears if in free trial mode)
- **Exit** Video Kiosk (This is configurable)

There are many options to protect access to the button bar and settings, which is important if your kiosk is accessible by the public (example, it’s a presentation on a tablet in a store). For more information, refer to the “Access Management” chapter.

### Settings Screen

To go to the Settings, press the Settings icon in the Button Bar. To disclose the settings, click the sub-screen title. There are an extensive number of settings that allow you to customize your Video

Kiosk. For complete information on the Settings Screen refer to the Configuring the Settings section of this document.



## Video Kiosk Settings Quick Reference Guide

Setting	Setting Location:
Android device's current <b>Time Zone</b>	For further information, refer to Appendix F
Android device's <b>Google Account</b>	For further information, refer to Appendix F
<b>Media Folder Location</b>	<p><b>To Check or To Change:</b> Button Bar &gt; Media Folder</p> <p><b>Default Location:</b> Standard folder Layout: /sdcard/videokiosk/media One folder Layout: /sdcard/videokiosk</p>
Cloud Download – <b>Cloud Source Folder Location</b>	<p><b>To Check:</b> Button Bar &gt; Settings &gt; Cloud Download &gt; Cloud download info</p> <p><b>To Change:</b> Button Bar &gt; Settings &gt; Cloud Download &gt; WebDAV folder or Button Bar &gt; Settings &gt; Cloud Download &gt; Nextcloud drive folder or Button Bar &gt; Settings &gt; Cloud Download &gt; Personal Cloud folder</p>
Cloud Download – <b>Local Folder Location</b>	<p><b>To Check:</b> Button Bar &gt; Settings &gt; Cloud Download &gt; Cloud download info</p> <p><b>To Change:</b> Button Bar &gt; Settings &gt; Cloud Download &gt; Path to local folder</p>
<b>Playback Order</b> Setting	<p><b>To Check or To Change:</b> Button Bar &gt; File and folder &gt; Sort by</p>
Playback Order <b>Playlist Folder location</b>	<p><b>To Check or To Change:</b> Button Bar &gt; File and folder &gt; Path to playlist folder</p>

## Using Settings XML File

If you need to configure many devices, you may prefer to create a `videokiosk.settings` XML file which you can copy to each device rather than having to manually configure the settings on every device.

Using the same Settings XML file on every device is a much faster way to configure the Video Kiosk settings when you're deploying a large number of devices.

For more information see: Appendix C Video Kiosk Settings XML section of this document.

# Features Overview

## Video Kiosk's default behavior

Video Kiosk is designed to be reliable, playing the contents of the media folder in a loop without any configuration. However, there are many additional features you can configure. If you have not configured otherwise, Video Kiosk will do the following:

### Display Mode

Default Display Mode

By default, Video Kiosk displays in Full Screen Mode. For more information, refer to [Understanding Screen Display Modes]

### Playback Control

Default Playback Order

If you have not configured a Playback Order, Video Kiosk will first look for a playlist and will play the contents of the Media Folder in the order of the first playlist it finds. If a playlist is not found, it will play the media folder contents in alphabetical order by full path name.

The Media Folder can have any number of sub-folders within it containing media. If you have not configured a playback sub-folder depth, Video Kiosk will play the contents down to 5 levels of sub-folders, ignoring sub-folders in lower depths. For more information about Playback Order, refer to [Controlling Playback].

### Display Control

Image Display Time

By default, Video Kiosk will display images for 30 seconds. You can increase or decrease this display time. For more information, refer to [Controlling the Display].

Web Page Display Behavior

By default, Video Kiosk will display web pages for 30 seconds. You can increase or decrease this display time. For more information, refer to [Controlling the Display].

### Device Control

Device Control – Dimming the screen

In order to play content, the device must be turned on.

When you launch Video Kiosk, it will play your content loop until you stop it. Therefore, the screen is not dimmed.

If you schedule playback instead, Video Kiosk will stop playing content at the scheduled time, dim the screen, and optionally display a clock.

If you use motion detection playback, Video Kiosk will stop playing content once it stops detecting motions, dim the screen, and optionally display a clock.

## Reliability Features

Reliability – Wake from sleep

Video Kiosk will automatically start playing content upon waking from sleep.

Reliability - Automatic Restart

Periodically restarting a device is a common way of addressing Android OS and hardware instability issues. Video Kiosk's default behavior is to restart once each night at midnight.

## Access Management

Access Management

By default, Video Kiosk is not password protected and no restrictions on device usage are in place. For more information about Access Management, refer to [Access Management, p. ].

# Video Kiosk Features

Once you have added your media and configured your Display Mode, your Video Kiosk is ready to go. However, there are many optional features available to customize your kiosk. Use this summary to decide how to enhance the reliability, useability and management of your Video Kiosk.

## Media Type Support

Video Kiosk supports playback of various Media Types – video, images and URL shortcuts. For more information, refer to [Working with Media Types].

- **Video**  
Supports playback of several video formats
- **Image**  
Supports display of several image formats
- **URL Shortcuts**  
Video Kiosk can open .url files (internet shortcuts), which are typically used to display live web content, but can be used to reference any URL on the internet or local to the device
- **Apps**  
Interactive Touch Screen Kiosks can open other Android Apps
- **Widgets**  
In Split Screen Display Mode, Widget Areas can play widgets

## Integrated Cloud Download

Video Kiosk can be updated through the cloud so that both the media that will be played and Video Kiosk functionality can be controlled from a remote location.

- **Remotely manage media files**  
Using a Cloud Service, you can remotely change the media that will be played on Video Kiosk.
- **Remote Configuration**  
Using a Cloud Service, you can remotely configure a device or many devices. This is especially useful if you are managing a fleet of devices.
- **Remote Access Management**  
Using a Cloud Service, you can remotely change the password for a device.

## Controlling Playback

Video Kiosk provides both simple ways and more sophisticated ways of controlling Playback Order and scheduling playback. For more information, refer to [Controlling Playback].

- **Playback Order Control**  
In addition to the default order, you can control the order media is played in using
  - a playlist
  - the file and folder order
  - random order
  - a Pick One folder
- **Playback Scheduling**  
You can schedule playback during particular times of the day using several methods
  - the Android Calendar on the device
  - a Google Calendar
  - a Schedule XML file
- **Playback by Device Location**  
You can control playback using the device location
- **Playback using Motion Detection**  
You can control playback using camera-based motion detection
- **Playback and Battery State Monitoring**  
Video Kiosk can monitor the battery state on the device and stop playing to conserve battery power
- **Changing the Playback Mode**  
By default, the Video Kiosk plays the next media item. However, you can change to play the previous, to pause after play or to loop play and require a swipe event to continue.

## Controlling the Display

In addition to just playing video or images, Video Kiosk supports Split Screen mode, where the playback screen is divided, allowing you to play video or images in the first area and run widgets in the others. As well, Video Kiosk supports Interactive mode where playback is determined by a user touching the screen or a keypad. But in addition, there are several other Display features available which control how your content looks on the device. For more information, refer to [Controlling Playback].

- **Black flash reduction**  
You can use textured video views and / or pause playback to reduce or eliminate the black flash between videos.
- **Screen Orientation Options**  
Pick from Landscape, Portrait, Reverse Landscape, Reverse Portrait
- **Image Display Options**  
Control how long each image is displayed, Fade images in and out, Scale or resize images
- **Video Display Options**  
Fade videos in and out, video resolution control for 4K display
- **Web Display Options**  
Preload URL files, Fade web pages in and out
- **Overlays**  
Video Kiosk supports transparent overlay files that are displayed in front of the currently displayed media. This feature is often used to overlay a company logo in the corner of the screen.
- **Background Images and Audio**  
Video Kiosk put a background image behind your control or play background audio.
- **Controlling Display using Meta data in file names**  
You can control playback when you include certain keywords in the media file's file name.
- **Legacy Display Support**  
Video Kiosk can stretch older video so it fits the new standard video size as well as reduce or eliminate the flicker at the beginning of videos on older Android devices.

## Controlling the Device

There are several features available which help you control the device on which your content is playing. For more information, refer to [Controlling the Device].

- **HDMI CEC support**  
Video Kiosk can send HDMI-CEC commands to keep the TV on and the correct input selected. (Supported on Android v5.1 and later.)
- **Keyboard and Remote Control**  
Video Kiosk supports some keys on keyboards and remote controls.
- **Touch Gesture Control**  
Video Kiosk supports using swipe gestures to select to the next or previous media item during playback.
- **Gamepad Control**  
Video Kiosk supports some buttons on gamepads.

## Reliability Features

Video Kiosk is designed to provide robust, reliable playback. By default, it handles media that can't be played, will restart if too many problems are encountered and seamlessly handles changes to the Videokiosk folder. There are some additional reliability features you can configure.

- **Behaviour on wake from sleep**

Video Kiosk will automatically start playing content upon waking from sleep. Use this feature to determine whether it will start playing from where it last ended in the play loop the current file or will start playing from the beginning of the play loop.

- **Automatic Restart**

Periodically restarting a device is a common way of addressing Android OS and hardware instability. Video Kiosk will automatically restart the device or restarts Video Kiosk at the specified intervals. This reduces the need to restart devices manually.

## Access Management

Because Video Kiosk is used in a variety of different kiosk environments, it may not be necessary for you to limit access to the device. For example, a wall-mounted TV used as a kiosk to play video at the gym would require less access management than an interactive kiosk in the hands of curious teenagers at a museum. Video Kiosk's Access Management features are designed to be flexible so that you can configure access at a level appropriate to the kiosk environment.

- **Setting the Home App**

In order to have Video Kiosk automatically restart when Android is restarted, set it as the Home App

- **Password Protection**

You require a password in order to make changes to the kiosk.

- **Disable the power button long press menu**

Disabling the power button long press prevents users from controlling the device using the power button long press menu.

- **Disable volume button**

Disable this to prevent users from changing the volume.

- **Prevent access to System UI**

This feature hides the system UI for a rooted Android v4 device and pins the Video Kiosk App (in Kiosk mode) for Android v5 and later devices where Video Kiosk is made the device owner. For an example of how to make Video Kiosk a device owner, refer to Appendix B – Making Video Kiosk the device owner at the end of this manual.

## Remote Management

Video Kiosk supports both sending status reports to a URL and receiving responses so you can integrate Video Kiosk remote devices into a management system.

# Understanding Screen Display Modes

Video Kiosk's playback screen supports three different modes in which to display your media:



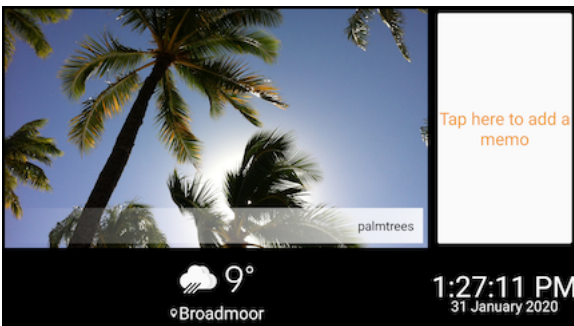
## Full Screen Mode

Full Screen Mode plays the content loop using the full playback screen. You can play **videos** or **images** or **URL-based content**, or any combination of the three.

You can also **control** what is displayed, for how long, in what order and even **schedule playback** for certain times of the day. Or, you could use **motion detection playback** to start the display when someone passes by, which attracts attention if it's an information kiosk in front of a museum exhibit or in a retail store.

Because **Full Screen is the default display mode**, all you need to do to use it is

- 1) Move your media on to your Android device
- 2) Launch Video Kiosk.



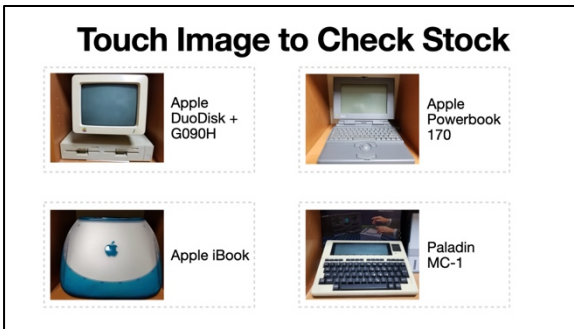
## Split Screen Mode

Split Screen Mode divides the screen into areas, so you can play videos or images or URL shortcuts, or any combination of the three in the main screen area and widgets or URL content in the other areas.

People passing by are always looking up to see the weather, a news feed or the time. Every time they do, they'll see the content you're displaying in the main screen area. You can also schedule playback or use motion detection

playback, control display times, add an overlay or a background.

For more information, refer to [How to Convert to Split Screen Display Mode].



## Touch Screen Interactive Kiosk Display Mode

Turn your Android device into an interactive Touch Screen (or mouse-controlled) kiosk, by providing a menu of options (in this case, the pictures) and configuring touchable hotspots that link to product web pages (the gray dashed-line boxes.) The hotspots borders are shown here for demonstration purposes only. You won't see them on the actual display.

For more information, refer to [How to Convert to Touch Screen Interactive Kiosk Display Mode].

# Working with Media Types

## Supported Media Formats

Video Kiosk will display videos, images or URL shortcuts on an Android TV, tablet or phone.

## Displaying Video

To play videos, put the video files in the Media folder.

- The video formats a given Android device supports are determined by the device manufacturer.
- If Video Kiosk encounters a video in a format not supported by the device, it will skip the video and will play the media next file. If the file is in a supported format, but unplayable for a different reason (for example an empty or corrupt file), Video Kiosk will skip it and play the next media file.

## Displaying Images

To play images, put the image files in the Media folder.

- Video Kiosk supports the following image file formats: .jpg, .jpeg, .gif, .png, .bmp, webp
- If your media folder includes still images, each one will be displayed for 30 seconds (this can be modified in the Settings).
- If Video Kiosk encounters an image in a format that is not supported, it will skip the file and will play the next media file.

## Using URL Shortcuts to display Content

Video Kiosk supports using URL shortcut files (.url files) to display content. URL shortcuts are files containing a link to any file that can be referenced by a URL, typically a web page. This feature is often used to display live web content, like an Instagram feed, but you can reference any URL on the web as well as locally stored html content and images.

- If your media folder includes .url files, the content will be displayed for the duration of time specified in the .url file. (The default is to display the web content for 30 seconds.)

- If Video Kiosk encounters an invalid URL, it will skip the file and will play the next media file. (A URL is considered invalid if the URL is incorrectly formed or if the page doesn't exist.)

## How to use URL Shortcuts

To use a URL shortcut to display content, follow these steps:

1. Create a .url File.  
In a Text Editor, create a URL file, with the extension “.url”.
  - For more information about URL files and to download a sample kit, refer to online Video Kiosk Tutorials
  - For file contents requirements, refer to the Video Kiosk URL Shortcut file Syntax Guide, below.
2. Move the .url files on to your device. For further information about moving files, refer to [Cloud Download]

## Video Kiosk URL Shortcut file (.url file) Syntax Guide

Syntax	Explanation	Required / Optional
[InternetShortcut]	Section identifier	Required
URL	Identifies the url value <b>Valid Values:</b> Any valid url address  <i>Example:</i> URL=http://www.burningthumb.com/	Required
TIMEOUT	Indicates how long, in seconds, Video Kiosk will display the web content. <b>Valid Values:</b> Default is 30 seconds Any valid floating point number  <i>Example:</i> TIMEOUT=45	Optional
CACHE-MODE	Indicates whether to cache the web content <b>Valid Values:</b> Default is no-cache no-cache – web content is not cached default – web content is cached  <i>Example:</i> CACHE-MODE=default	Optional
LAYER-TYPE	Indicates where to render the web content <b>Valid Values:</b> Default is hardware hardware – web content is rendered on the hardware accelerated layer software – web content is rendered on the software layer	Optional

	<i>Example:</i> LAYER-TYPE=software	
USER-AGENT	Allows you to specify the user agent sent to the web server.  <i>Example:</i> USER-AGENT=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5) AppleWebKit/603.2.4 (KHTML, like Gecko) Version/10.1.1 Safari/603.2.4	Optional
MUTE-ON-PRELOAD	If the web page contains audio content, the content volume can be muted during background pre-loading of the content <b>Valid Values:</b> Default is false true – the volume will be muted false – the volume will not be muted  <i>Example:</i> MUTE-ON-PRELOAD=true	Optional
SHOW-AFTER-PRELOAD	Normally the web page will be displayed when the previous item finishes playing. You can specify that the web page should be shown as soon as the preloading is complete. <b>Valid Values:</b> Default is false true – the page will be displayed as soon as the preload completes false – the page will be displayed when the previous item is finished playing  <i>Example:</i> SHOW-AFTER-PRELOAD=true	Optional
SCALE-IMAGE	If the URL is an image URL and you set this option to true, Video Kiosk will scale the image to full screen.  <b>Valid Values:</b> Default is false true – scale the image to full screen false – position the image based on the default for the WebView. This varies based on the version of Android being used  <i>Example:</i> SCALE-IMAGE=true	Optional

## Examples: URL Shortcut Files

The following example shows the content of a .url file that instructs Video Kiosk to play the BurningThumb Studios home page for 30 seconds without caching the web page content:

```
[InternetShortcut]
URL= http://www.burningthumb.com/
TIMEOUT=30
LAYER-TYPE=software
CACHE-MODE=no-cache
```

The following example shows the content of a .url file that instructs Video Kiosk to auto play a YouTube video for 138 seconds while caching the web page content. Notice that a desktop (Mac OS X) USER-AGENT is specified so that the video will auto play even though the real user agent is on a mobile device which would result in auto play not working:

```
[InternetShortcut]
URL=https://www.youtube.com/embed/IYQWiQJ4IE?rel=0&autoplay=1
TIMEOUT=138
LAYER-TYPE=hardware
CACHE-MODE=default
USER-AGENT=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5) AppleWebKit/603.2.4 (KHTML, like Gecko) Version/10.1.1 Safari/603.2.4
```

The following example displays an image and scales it to use the full screen:

```
[InternetShortcut]
URL=https://media-beta.wsbtv.com/photo/MAP/27381361/27381361\_Position3\_660\_500.JPG
TIMEOUT=30
SCALE-IMAGE=true
```

For more information about URL files and to download a sample kit, refer to the tutorials at:

<http://burningthumb.com/apps/video-kiosk/tutorials/>

# Cloud Download

There are several ways you can move your files from your computer to Video Kiosk:

- Connect your Android device to your computer with USB cable and copy them over using the Android File Transfer app (see Appendix F for details)
- Use external storage like a USB stick or SD card
- Have Video Kiosk download the files automatically using the Cloud Download feature.

Video Kiosk's integrated Cloud Download features allows you remotely update both the media that will be played as well as Video Kiosk's Configurations files using either WebDAV, Nextcloud, or our Personal Cloud PHP script on your web server as your cloud service (integrated Google Drive support has been deprecated).

**Please note that this feature will download from the Cloud folder to the device(s) folder, but will not upload from the device to the Cloud.**

**Ensure that you always modify the content in the Cloud folder, not the local folder on a device.**

For example, if you add an image to a folder, named videokiosk for this example, on the cloud service, the folder on all the devices will be updated with the image. However, if you add an image to a folder, named videokiosk in this example, on one of the devices, that change will not be uploaded to the Cloud or to any of the other devices. Instead, that image file will be deleted the next time the devices are updated with the contents of the Cloud folder.

If you need Google Drive support, or you prefer for other reasons, you can use a third-party cloud download App like Autosync Dropbox or Autosync Google Drive. The process of configuring a third-party cloud download App would be similar to the procedures below. However, we do not provide support for configuring third-party downloading Apps. If you're going to use a third-party downloading app, we recommend that you configure it in download only (sometimes called mirror) mode.

## How to Configure and Use Cloud Download Update

**Internet Access is required on your Android device to configure and use Cloud Download Update.**

To set up the Cloud Download Update feature, follow these steps:

1. If you haven't already done so, **create a Cloud Source folder**.  
e.g. WebDAV service > Video Kiosks > Phone Video Kiosk > videokiosk

(Management Hint: On your computer or cloud drive, put your videokiosk folder inside another folder with a meaningful name to help identify it, especially if you're running more than one Video Kiosk.)

2. On each Android device, **configure the settings with the Cloud Source Folder location.**

Video Kiosk > Settings > Cloud Download > WebDAV folder

OR

Video Kiosk > Settings > Cloud Download > Nextcloud folder

OR

Video Kiosk > Settings > Cloud Download > Personal Cloud folder

Use the file navigation to select Cloud Source Folder. It will be displayed as the Current folder at the top of the screen (For this example, "Current folder: videokiosk")

Tap "Choose"

3. If you haven't already done so, **create a local folder** on each Android device.

Launch Video Kiosk

Video Kiosk > Button Bar > Folder icon

From the Ellipse menu (:), select "Create Folder"

Enter the folder name and tap "Confirm"

4. On each Android device, **Configure the settings with the device's local Folder location.**

Video Kiosk > Settings > Cloud Download > Path to Local folder

Use the file navigation to select Cloud Source Folder. It will be displayed at the top of the screen (For this example, "/storage/.../videokiosk")

Navigate **into** the folder you created above

Tap "Choose"

5. Optionally, change the **Download Interval**. Default is every 1 hour.

Video Kiosk > Settings > Cloud Download > Download Interval

6. Before you enable downloading, **verify your settings**. Before you continue, verify that the Source and Target locations are correct.

Video Kiosk > Settings > Cloud Download > Cloud drive info

To return to the previous screen, press OK.

7. **Enable Downloading.**

Video Kiosk > Settings > Cloud Download > Enable downloading

To download to the device(s), just move the files you want to update to the folder on your Cloud Service.

Video Kiosk will check for updates on the Cloud Service periodically, as set in the Download Interval Setting (Video Kiosk > Settings > Cloud Download > Download interval) and update the devices with the updates you have made, as scheduled.

**If you want to download right now,**

- Navigate back to the main Video Kiosk screen by pressing the back button, and
- Double tap the Video Kiosk screen.

Download will begin immediately, and a toast message will be displayed indicating that is the case.

For more information on how to use your web site and our Personal Cloud PHP script, refer to **Appendix G – WebDAV**

**WebDAV (Web Distributed Authoring and Versioning)** is an industry standard that you can use to host files on any web server. A large number of cloud services support WebDAV and you can create your own WebDAV server if you own a web site.

For example, cpanel users can create a Web Disk which is accessible using WebDAV.

Your web hosting provider can help you setup a WebDAV server and Video Kiosk can be configured to login to that server and download your files to all of your devices.

If you're planning on using WebDAV, we recommend you create an account on your WebDAV server specifically for Video Kiosk, and if possible, configure it in read-only mode.

You can find complete information on configuring WebDAV using cpanel here:

<https://cpanel.net/blog/tips-and-tricks/how-to-configure-and-manage-webdav-web-disks-with-cpanel/>

**Appendix H – Personal Cloud**

# Controlling Playback

With video Kiosk, you can control both the **order the Content Loop is played** (Playback Order) as well as **when the Content Loop is played** (Playback Scheduling). Video Kiosk provides a number of options for controlling both of these.

## Default Playback Behavior

If you have not configured a Playback Order, Video Kiosk will play the contents of the Media Folder (the Content Loop) in the order of either the first playlist it finds or, if a playlist is not found, the media folder contents in alphabetical order by the full path name of each file..

If you have not configured a time for playback, Video Kiosk will play the contents of the Content Loop whenever Video Kiosk is running.

## Controlling Playback Order

You can control the playback order of the contents of the media folder using the following methods:

- The filename or filename and path order
- Random Order
- A .pickone folder(s)
- A playlist

A playlist will take precedence over the other methods, so if you select a playlist, Video Kiosk will use the order contained in the .m3u8 playlist file.

## Controlling playback order with Filename

When using either **Path** sort order or **File name excluding path** sort order, Video Kiosk will play the media in alphabetic order (UTF8 order), which sorts numbers (0-9) before letters (a-z).

For more information on UTF8, refer to UTF-8 - Wikipedia, The Free Encyclopedia at <https://en.wikipedia.org/w/index.php?title=UTF-8&oldid=683120706>

If you want to control sort order, we recommend you prepend file names with a 2-4 digit number for Video Kiosk to sort by.

### *Example:*

In this example, there are 6 media assets, 3 videos clips from competitions and 3 images - 1 upcoming event poster image, 1 ad spot for the Pro Shop image and 1 schedule image. To

intersperse the videos between the images, prepend the file names with number to control the sort order and choose filename as the sort order. The filenames would look like this:

001 pro-shop-spot.png  
002 brandon-fight-clip.mp4  
003 mike-boxing-102.png  
004 david-fight-clip.mp4  
005 pma-schedule-fall-2015.png  
006 robert-fight-clip.mp4

The prepended numbers make the sort order easy to understand and easy to see.

## Random order playback

To use random order playback, follow these steps:

1. Configure Video Kiosk with a sort order.  
Go to **Video Kiosk Settings > File and Folder > Sort by** and set the order to Random shuffle

## Controlling playback order with a Pick One folder

Video Kiosk recognizes the special folder extension, **.pickone**, and will select just one of the files in that (sub-)folder (in a round robin fashion) each time it plays the content loop.

To use a Pick One folder, follow these steps:

1. Create a folder (or number of folders), using the extension, **.pickone**, and put the media you want to play in the folder(s).
2. Put the folders in the media folder.

Video Kiosk will select one file to play from each **.pickone** folder each time it plays the media loop.

## Controlling playback order using a Playlist

Video Kiosk supports using playlists (m3u8 files) to control the playback order of the content loop.

### How to use a Playlist

To use a Playlist to control the playback order, follow these steps:

1. **Create a Playlist file.** Create a Plain Text playlist file, with the extension “.m3u8”.
  - Because the file must be a “Plain Text” encoded file, you can’t use a Word Processor to create a Playlist file.
  - You can do this using a Playlist file creation app.
  - If you are not using an App to create your playlist, the easiest way to create a Playlist File is to download the Playlist Sample Kit, and edit a sample playlist using a Text Editor (Notepad on Windows, TextEdit on Mac). To download the sample kit, refer to the

tutorials at:

<http://burningthumb.com/apps/video-kiosk/tutorials/>

- For more information on m3u8 files and m3u8 file syntax, refer to M3U - Wikipedia, The Free Encyclopedia at <https://en.wikipedia.org/w/index.php?title=M3U>
  - For file contents requirements, refer to the Syntax Guide, below.
2. **Move the file.** Move the file to “videokiosk/playlist/” on the device, either using the Cloud Download Feature or using a USB and Android File Transfer.
  3. **Configure the Path.** If you are using a Custom Folder Layout, configure Video Kiosk with the location of the Playlist file. (If you are using the Standard Folder Layout or One Folder Layout, this step is not required.)  
Go to **Video Kiosk Settings > File and Folder > Path to Playlist folder** and set the path to the playlist folder.
  4. Select the File.  
Go to **Video Kiosk Settings > File and Folder > Sort by** and select the playlist to use from the list to use. If you don’t set this, Video Kiosk will play the first playlist it finds in the Playlist folder.

## Video Kiosk Playlist Syntax Guide

#EXTM3U #EXT-X-PLAYLIST-TYPE:VOD #EXT-X-VERSION: #EXT-X-MEDIA-SEQUENCE:0		Required
#EXTINF:99.999, (filename.ext)	Identifies the track (“#EXTINF:99,” required for each entry. Optionally use the rest of the line to identify the track) Specify the Filename. Either specify the Full Path name on the device or specify the relative path from you Video Kiosk Media Folder location (Usually videokiosk/media/)	Required for each track
#EXT-X-ENDLIST	Indicates end of file	Required

## Examples: Playlist Files

### Example 1

Here is an example playlist.m3u8 file that plays 3 files, starting with “christmas-lunch-specials-2017.mpg” and ending with “snow.mp4”. These three media files are in a sub-folder, called “Christmas” within the videokiosk media folder  
(file location for “snow.mp4” is videokiosk/media/Christmas/snow.mp4)

```

#EXTM3U
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0

#EXTINF:305.0,
Christmas/christmas-lunch-specials-2017.mpg

#EXTINF:303.0,
Christmas/social-events-2017.mpg

#EXTINF:303.0,
Christmas/snow.mp4

#EXT-X-ENDLIST

```

### Example 2

Here is an example playlist.m3u8 file that plays 3 files, starting stored in the videokiosk media folder

```

#EXTM3U
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
|
#EXTINF:305.0,
02-bts-vka-mediapack-drive-download-banner.jpeg

#EXTINF:305.0,
04-bts-vka-mediapack-video-kiosk-banner.jpeg

#EXTINF:305.0,
06-bts-vka-mediapack-shredit-android-banner.jpeg

#EXT-X-ENDLIST

```

Note: Currently the #EXTINFO values are ignored, but the .m3u8 parser expects them to be present.

## Filename versus Filename with Path Options

You can control whether the sort includes the file path using the Video Kiosk **File and folder > Sort by** settings.

When **Path** sort order or **File name excluding path** sort is selected, Video Kiosk will play the media in alphabetic order (UTF8 order) by either full path or just the file name component of the path. For more information on UTF8, refer to UTF-8 - Wikipedia, The Free Encyclopedia at <https://en.wikipedia.org/w/index.php?title=UTF-8&oldid=683120706>

For information on Settings, refer to Configuring the Settings section of this document.

To control playback order using the settings, follow these steps:

1. Configure Video Kiosk with a sort order. Go to **Video Kiosk Settings > File and Folder > Sort by** and set the set order. The default is **Path of first playlist found**.

#### Options:

Path or first playlist found – Sorts by path or by the first playlist found (if there is one)

Path – Sorts by full path name

File name excluding path – Sorts by file name, excluding path  
Sort Order Considerations

Here are some considerations when configuring a sort order.

## Restricting File Types

Android has a list of built-in supported video formats. The manufacturer of your device may support other formats. By default, Video Kiosk will attempt to play video in a format not on the built-in supported list and skip the video if it can't be played.

To restrict playback to just built-in supported video formats,

1. Go to **Video Kiosk Settings > File and folder > Limit playback to documented formats** and check the checkbox.

## Displaying Playback Error Messages

You can troubleshoot video playback by displaying a message each time a video doesn't play.

To enable Display Playback Error Message,

1. Go to **Video Kiosk Settings > File and folder > Show playback errors** and check the checkbox.  
Video Kiosk will then display a brief message when a media item cannot be played. Once you have identified and corrected the problem, we recommend you turn off this setting before deploying production devices.

## Limiting the sub-folder depth

When controlling playback using the File and folders Settings or a .pickone folder, make sure the folder depth is set high enough to play all the sub-folders. When Video Kiosk is searching a media folder, the default limit to how many sub-folders depths it will traverse is set to 5. If you have a lot of nested sub-folders, you can increase the limit.

1. Pick the limit of the depth of nested sub-folder that will be played. Go to **Video Kiosk Settings > File and folder > Sub-folder depth** and select the depth.

## Controlling Playback Time using a Schedule

In addition to controlling Content Loop order, you can control the time at which the Content Loop is displayed using a schedule. You can use any one of the following methods:

- **Using the Android calendar on each device.** Since Android Calendar is on the Android device, you'll need to enter the schedule events on each of your devices. To modify the schedule, you'll need to enter the new schedule events on each device. This requires physical access to each device but doesn't require internet access.

- **Using Google calendar.** You can have many Android devices share the same Google Calendar. You enter the schedule events on a Google Calendar and then configure each device to use that calendar. To modify the schedule, you just need to enter the new schedule events on the Google calendar and all the devices will automatically switch to the new schedule. This doesn't require physical access to each device but does require that each device has internet access.
- **Using an XML schedule file on each device.** To use this option, you create an XML Schedule file and deploy it on each device. You can have many Android devices share the same XML Schedule. To modify the schedule, update the XML Schedule file and either copy it to each device using a USB Cable (requires physical access to each device) or update using Video Kiosk's Cloud Download feature (requires that each device has internet access).

## Deciding on a Schedule option

All of the scheduling options will involve some initial set up time. In addition, there will be some work involved in making modifications to the schedule. When deciding on which schedule option to use, choose an option that will make it easy to make modifications, especially if the schedule changes often, you have a lot of devices or the devices are hard to access because they're installed high on a wall or in another city.

For example, if you plan to change the schedule every two weeks and your devices are installed high on the wall in theatres across the country, choose a scheduling option that doesn't require physical access to each device. Here are some things to consider:

- Do the devices have internet access?
- How accessible are the devices – are they difficult to get to because they are mounted overhead on a wall or in another city?
- How many devices are being updated?
- How often will the schedule change?
- Is the schedule the same every week or different?

A Calendar schedule can be scheduled on a day-to-day basis. An XML Schedule file can be scheduled on a week-to-week basis.

Example: You have your Saturday Schedule, but want to add an event for just this weekend. With a Calendar Schedule, you would just add an event to the schedule for the one day. With an XML Schedule file, you would change the Saturday schedule and deploy it just before that Saturday and then change it back again for subsequent Saturdays.

Which scheduling option best meets your requirements will depend on your configuration. Here are the considerations for each option.

Consider this:	Android Calendar	Google Calendar	XML Schedule File copied over with USB cable	XML Schedule File updated using Cloud Download feature
Do the Android Devices have internet Access?	Not required	Required	Not required	Required

Consider this:	Android Calendar	Google Calendar	XML Schedule File copied over with USB cable	XML Schedule File updated using Cloud Download feature
If physical access to the device is required for update, how accessible are the devices?	Physical Access required for update	Physical Access not required for update	Physical Access required for update	Physical Access not required for update
How many devices are being updated?	Update devices one at a time	Update all devices simultaneously	Update devices one at a time	Update all devices simultaneously
How often will the schedule change?	Physical Access required for update - frequent schedule changes will be labor-intensive to deploy	Physical Access not required for update - frequent schedule changes will be easy to deploy	Physical Access required for update - frequent schedule changes will be labor-intensive to deploy	Physical Access not required for update - frequent schedule changes will be easy to deploy
Is the schedule the same every week or different?	Can easily handle scheduling exceptions	Can easily handle scheduling exceptions	Designed to handle schedules that are consistent week by week	Designed to handle schedules that are consistent week by week

## Using Android Calendar to control the schedule

Video Kiosk will execute commands based on the **time** and **description** of the calendar entry in the calendar you have selected in the settings. To use Android Calendar to schedule playback, follow these steps:

1. **Create a Calendar.** In Android Calendar on the device, create a calendar.
2. **Add events.** Add playback events using the start and end times in the Calendar to control when your content is played (or not played, depending on the Display State setting.)  
Optionally, you can control device behavior by adding commands to the description field of the event. For more information, refer to the **Video Kiosk Calendar Command Reference Guide** section of this document.
3. **Configure with Location.** Configure Video Kiosk with the location of the schedule.  
Go to **Video Kiosk Settings > Schedule > Use a calendar or schedule** and select the Android Calendar as the schedule location. Video Kiosk will display all Android Calendars on the device in this list.
4. **Behavior when No Event.** Choose the device behavior when no schedule event occurs. The default is to display a blank screen.  
To play the clock instead, go to **Video Kiosk Settings > Schedule > Display Clock** and set to on. You can change the clock color using the **Video Kiosk Settings > Schedule > Clock Color** setting. You can change the clock format (12 hour AM/PM versus 24 hour) in the System Settings for the device.

5. **Set Schedule Setting Display State.** Choose whether the schedule event controls the display by setting the Schedule Setting **Display state**.  
**Always on** - the schedule events do not turn the display off  
**On for events, otherwise off** - the schedule event will turn the display on  
**Off for events, otherwise on**, the schedule event will turn the display off.
6. **Check Schedule Frequency.** Choose how often Video Kiosk checks for changes to the schedule. The default is every 30 seconds. You can change the clock color using the **Video Kiosk Settings > Timers > Schedule Check seconds** setting.
7. **Device Time.** Ensure that the device's time is set to the correct time and correct time zone. (This may be set automatically, depending on the device. For more information, refer to the Owner's manual that came with the device.)

## Using Google Calendar to control the schedule

Video Kiosk will execute commands based on the **time** and **description** of the calendar entry in the calendar you have selected in the settings. If you're using Google Calendar to schedule playback, you'll need to ensure the device has access to the Internet.

### How to use a Google Calendar Schedule

To use Google Calendar to schedule playback, follow these steps:

1. **Create Calendar.**

In Google Calendar, create a calendar.

2. **Add Events.**

Add playback events using the start and end times in the Calendar to control when your content is played (or not played, depending on the Display State setting.)

Play Vintage TV SAVE

Dec 5, 2017 10:00am to 6:00pm Dec 5, 2017 TIME ZONE

All day Weekly on Tuesday, Wednesday, Thursday, Friday, Saturday ▾

3. **Add Device Control commands.**

Optionally, add device control commands to the calendar event description field, if required. For more information, refer to the **Video Kiosk Calendar Command Reference Guide** section of this document.

4. **Configure Video Kiosk with schedule location.**

First, on the device, logon to the Google Account you used when creating the Google Calendar. This will notify Video Kiosk of the existence of the Google Calendar and add it to the Video Kiosk list of schedules.

Next, configure Video Kiosk with the location of the schedule.

Go to

**Video Kiosk Settings > Schedule > Use a calendar or schedule**

and select the Google Calendar as the schedule location.

5. **Behavior when No Event.**

Choose the device behavior when no schedule event occurs.

The default is to display a blank screen.

To play the clock instead, go to **Video Kiosk Settings > Schedule > Display Clock** and set to on. You can change the clock color using the **Video Kiosk Settings > Schedule > Clock Color** setting. You can change the clock format (12 hour AM/PM versus 24 hour) in the System Settings for the device.

6. **Set Schedule Setting Display State.** Choose whether the schedule event controls the display by setting the Schedule Setting **Display state**.

**Always on** - the schedule events do not turn the display off

**On for events, otherwise off** - the schedule event will turn the display on

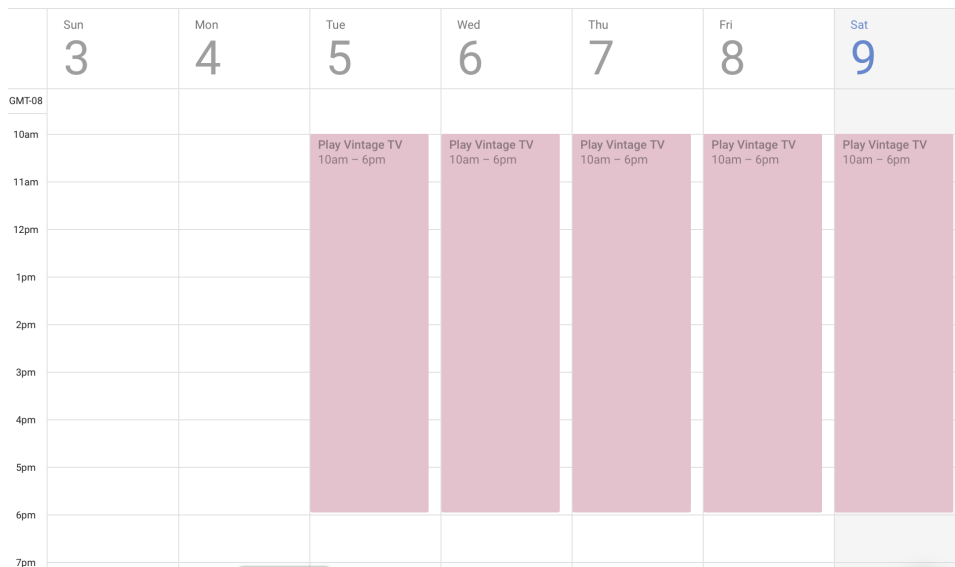
**Off for events, otherwise on**, the schedule event will turn the display off.

7. **Check Schedule Frequency.** Choose how often Video Kiosk checks for changes to the schedule. The default is every 30 seconds. You can change the clock color using the **Video Kiosk Settings > Timers > Schedule Check seconds** setting.

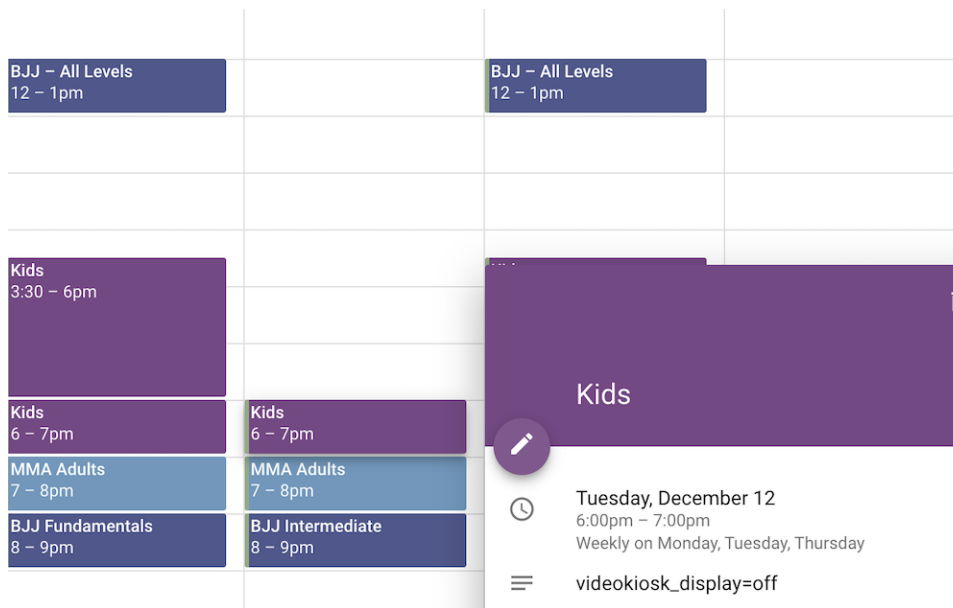
8. **Device Time.** Ensure that the device’s time is set to the correct time and correct time zone. (This may be set automatically, depending on the device. For more information, refer to the Owner’s manual that came with the device.)

### Video Kiosk Google Calendar Schedule examples

Here is an example of a Google calendar schedule used by a retail store, displaying the content loop during the store’s opening hours, turning the display on at 1000 and off at 1800 from Monday to Saturday.



Here is an example of a Google schedule used by a dojo, displaying the content loop when classes are running, except during the kid’s classes. To turn content display off for this one event, the command “videokiosk\_display=off” is added to the event’s description.



## Video Kiosk Calendar Command Reference Guide

Android Calendar or Google Calendar may also be used to control Video Kiosk's behavior. Video Kiosk will check the calendar and execute commands based on the **time** and **description** of the calendar entry. By default, at the time of the calendar event, Video Kiosk will start playing the loop according to how the Playback Order is set and stop playing at the end of the calendar event. To change this behaviour, include these commands in the description of the calendar entry.

Command	Arguments	Description
videokiosk_display_state	{alwayson, eventon, eventoff}	Set the display state to always on (scheduled events will not change the display), On for events, otherwise off (scheduled events will turn the display on, otherwise the display will be off), or Off for events, otherwise on (scheduled events will turn the display off, otherwise the display will be on). Changing this will update the settings.  If the <b>videokiosk_display</b> command is present, it overrides this setting for that specific event.
videokiosk_path	/path/to/folder	Set the playback path used by Video Kiosk to the <b>Media folder</b> on the device. Use the path to the Media Folder, not a file in the Media Folder.  This command is equivalent to changing the <b>Media folder</b> in the App by clicking the <b>folder</b> icon in the button bar. Changing this will update the settings.  If you want to temporarily change the path, just for this event, you need to create 2 events. The first event should set the temporary path and it should be followed immediately by the second event that resets to the original path.
videokiosk_display	{on,off}	Turn the display on or off for this one event. This overrides the display state but only for this one event. For

Command	Arguments	Description
		example, if the default is display on, you can turn display off for specific events using this command.

### Examples:

- To turn content display off for just one event add this line to the event description:  
videokiosk\_display=off
- To change the path to the Media Folder, add this line to the event description:  
(Since the path depends on the file system of the device, check the path with a file browser on your device prior to using this command.)  
videokiosk\_path=/mnt/sdcard/Current Schedule

## Using an XML file to control the schedule

You can use an .XML file to control the schedule.

### How to use an XML Schedule file

To use an XML schedule file to schedule playback, follow these steps:

1. **Create the File.** In a Text Editor, create an XML schedule file. When you have finished editing the file, change the file extension to “.schedule” so Video Kiosk will recognize it as a Schedule file. For more information, refer to the **Video Kiosk XML Schedule File Syntax Reference Guide** below.
2. **Move the File.** Move the file onto the device(s) to “videokiosk/schedule/”
3. **Configure the Path.** If you are using a Custom Folder Layout, configure Video Kiosk with the location of the location file(s). (If you are using the Standard Folder Layout or One Folder Layout, this step is not required.)  
Go to **Video Kiosk Settings > Schedule > Path to schedule folder** and set the path to the schedule folder.
4. **Select the File.**  
Go to **Video Kiosk Settings > Schedule > Use a calendar of schedule** and select the schedule to use.
5. In **Video Kiosk > Settings > File and folder > Sort by**, choose **Path or first playlist found**. Scheduled playback won't work with the **Sort by** settings (**Path, File name excluding path, Random shuffle**) enabled.
6. **Behavior when No Event.**  
Choose the device behavior when no schedule event occurs.  
The default is to display a blank screen.  
To play the clock instead, go to **Video Kiosk Settings > Schedule > Display Clock** and set to on. You can change the clock color using the **Video Kiosk Settings > Schedule > Clock Color** setting. You can change the clock format (12 hour AM/PM versus 24 hour) in the System Settings for the device.
7. **Set Schedule Setting Display State.** Choose whether the schedule event controls the display by setting the Schedule Setting **Display state**.  
**Always on** - the schedule events do not turn the display off

**On for events, otherwise off** - the schedule event will turn the display on  
**Off for events, otherwise on**, the schedule event will turn the display off.

8. **Check Schedule Frequency.** Choose how often Video Kiosk checks for changes to the schedule. The default is every 30 seconds. You can change the clock color using the **Video Kiosk Settings > Timers > Schedule Check seconds** setting.
9. **Device Time.** Ensure that the device’s time is set to the correct time and correct time zone. (This may be set automatically, depending on the device. For more information, refer to the Owner’s manual that came with the device.)

## Video Kiosk XML Schedule file Syntax Guide

<events> </events>			A list of all playback events	Required
	<event> </event>		A single playback event	Required for event
		<week> </week>	Events occurs on multiple days of the week. Define days using the string “SMTWTFS”, using the letter to include the day and a “-” to skip the day. Use all 7 characters or the event will be ignored.	Use either <week> or <dayofweek> for each event, not both
		<dayofweek> </dayofweek>	Event occurs for a single day <b>Valid Values:</b> Sun Mon Tue Wed Thu Fri Sat	Use either <week> or <dayofweek> for each event, not both
		<starttime> </starttime>	The time the event starts (24hour clock, format “99:99”)	Required for event
		<endtime> </endtime>	The time the event ends (24hour clock, format “99:99”)	Required for event
		<playlist> </playlist>	A playlist to use for this event	Optional
		<path> </path>	A path to use for this event	Optional

## Video Kiosk .schedule XML file examples

The week tag is used to define the same event occurring on multiple days of the week, using a string of the form SMTWTFS, where each day of the week uses the first letter of the days name. If you want the event to occur on a day, use the letter for the day’s name. If you want to skip a day, use a “-” instead of the first letter of the day’s name. It’s important that the string always contains all 7 characters. If you use any other number of characters, the event will be ignored.

For example,

- An event that occurs Monday through Friday would use the string “-MTWTF-”
- An event that occurs only on the weekend would use the string “S-----S”
- An event that occurs on Monday, Wednesday and Friday would use the string “-M-W-F-”

Here is an example of an XML file that contains two events, the first displaying the morning playlist weekday mornings from 06:00 to 11:59 and the second displaying the afternoon playlist every afternoon from 12:00 to 20:59:

```
<?xml version="1.0" encoding="UTF-8"?>
<events>
  <event>
    <week>-MTWTF-</week>
    <starttime>06:00</starttime>
    <endtime>11:59</endtime>
    <playlist>morning.m3u8</playlist>
  </event>
  <event>
    <week>SMTWTFS</week>
    <starttime>12:00</starttime>
    <endtime>20:59</endtime>
    <playlist>afternoon.m3u8</playlist>
  </event>
</events>
```

The dayofweek tag is used to define an event occurring on one day of the week. The values to use for the days of the week are:

Sun,  
Mon  
Tue  
Wed  
Thu  
Fri  
Sat

For example, an event that occurs each Monday would use the string "Mon"

Here is an example of an XML file that plays the morning playlist on Monday mornings and the afternoon playlist on Thursday afternoons, except for one hour starting at 4 o'clock.

- On Monday mornings, display the morning playlist from 0600 to 1159
- On Thursday afternoons, display the afternoon playlist from 1200 to 1600
- On Thursday afternoons, display the afternoon playlist from 1700 to 20:59

```
<?xml version="1.0" encoding="UTF-8"?>
<events>
  <event>
    <dayofweek>Mon</dayofweek>
    <starttime>06:00</starttime>
    <endtime>11:59</endtime>
    <playlist>morning.m3u8</playlist>
  </event>
  <event>
    <dayofweek>Thu</dayofweek>
    <starttime>12:00</starttime>
    <endtime>16:00</endtime>
    <playlist>afternoon.m3u8</playlist>
```

```
</event>
<event>
  <dayofweek>Thu</dayofweek>
  <starttime>17:00</starttime>
  <endtime>20:59</endtime>
  <playlist>afternoon.m3u8</playlist>
</event>
</events>
```

For more information about XML Schedule files and to download a sample kit, refer to the tutorials at:

<http://burningthumb.com/apps/video-kiosk/tutorials/>

## Controlling Playback using device location

You can control playback using the device location and defined by Geofences. Geofences can be specified in two ways:

- Video Kiosk .location XML files
- GeoJSON File with Video Kiosk Extensions

*It's important to consider overlapping Geofences, in the case where Geofences overlap, and a device is located in the overlapping area, Video Kiosk will choose the playlist from the smaller of the two areas. If the two areas are the same size, Video Kiosk will choose the playlist from the first area.*

### Location XML file

Video Kiosk will select a playlist based on the location of the device and the Geofences defined in a .location XML file:

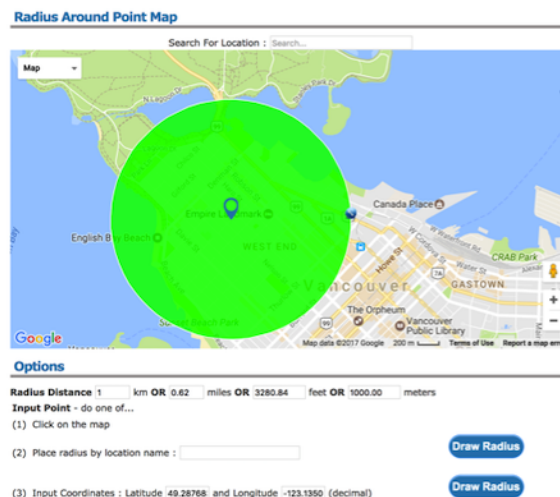
1. **Create the File.** In a Text Editor, create a Location XML file(s) and use them to control playback. For more information, refer to the Video Kiosk .location file Syntax Guide below.
2. **Move the File.** Move the file(s) onto the device or synchronize it using a cloud service. (Move to "videokiosk/location/")
3. **Configure the Path.** If you are using a Custom Folder Layout, configure Video Kiosk with the location of the location file(s). (If you are using the Standard Folder Layout or One Folder Layout, this step is not required.)  
Go to **Video Kiosk Settings > Location > Path to location folder** and set the path to the location folder.
4. **Select the File.** Select Video Kiosk location file that defines the Geofences you want to use. Go to **Video Kiosk Settings > Location > Use Geofence locations** and select the location file to use.

## Video Kiosk .location file Syntax Guide

<fences> </fences>					A list of all Geofences	Required
	<fence> </fence>				A single Geofence	Required
		<name> </name>			The name of the Geofence	Required
		<circle> </circle>			A circular Geofence. Defines a Geofence with a center and a radius. <i>Note you should select a circle OR a polygon but not both.</i>	Optional
			<lng> </lng>		The longitude of the center point of the circular Geofence	Required for circle
			<lat> </lat>		The latitude of the center point of the circular Geofence	Required for circle
			<radius> </radius>		The radius, in meters, of the circular Geofence	Required for circle
		<polygon> </polygon>			Defines a Geofences as a series of points. Each polygon must have a minimum of 3 points. . <i>Note you should select a circle OR a polygon but not both.</i>	Optional
			<point> </point>		One point in a polygon	
				<lng> </lng>	The longitude of the a point of the polygon Geofence	Required for point
				<lat> </lat>	The latitude of a point of the polygon Geofence	Required for point
		<playlist> </playlist>			A playlist to use when the device is geographically located in this Geofence	Required

## Location XML Example

Below is an example of the XML required to create a 1km circular geofence in the West End of Vancouver, BC, Canada.



```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<fences>
  <fence>
    <name>WestEnd</name>
    <lat>49.28768</lat>
    <lng>-123.1350</lng>
    <radius>1000</radius>
    <playlist>westend.m3u8</playlist>
  </fence>
</fences>
```

## Location GeoJSON file

Video Kiosk will select a playlist based on the location of the device and the Geofences defined in a .geojson JSON file:

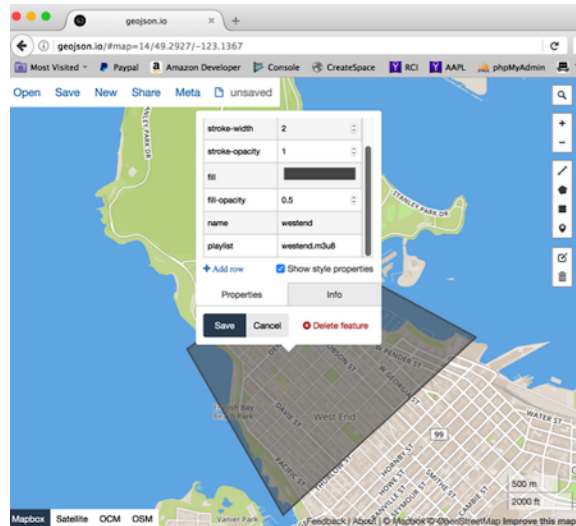
1. **Create the File.** On the web site [geojson.io](https://geojson.io), create the map that you wish to use. Add the extra, required, fields name and playlist to each polygon on your map. For more information, refer to the Video Kiosk GeoJSON Syntax Reference Guide below. Save the file(s) as GeoJSON (.geojson).
2. **Move the File.** Move the file(s) onto the device or synchronize it using a cloud service.
3. **Configure the Path.** If you are using a Custom Folder Layout, configure Video Kiosk with the location of the location file(s). (If you are using the Standard Folder Layout or One Folder Layout, this step is not required.)  
Go to **Video Kiosk Settings > Location > Path to location folder** and set the path to the location folder.
4. **Select the File.** Select Video Kiosk location file that defines the Geofences you want to use. Go to **Video Kiosk Settings > Location > Use Geofence locations** and select the location file to use.

## Video Kiosk .geojson file Syntax Guide

Since JSON is not as human readable as XML, it is strongly recommended that you use the web site [geojson.io](https://geojson.io) to create and edit your .geojson files. In addition to the standard geojson generated when a polygon is created on this site you need to add the fields **name** and **playlist**:

- **name** is a unique string that is used to name the polygon.
- **playlist** is the file name (not the full path just the filename part of the path) of the playlist that you want to use.

Below is an example of the geojson created at [geojson.io](https://geojson.io) with the extra fields added.



```

{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "stroke": "#555555",
        "stroke-width": 2,
        "stroke-opacity": 1,
        "fill": "#555555",
        "fill-opacity": 0.5,
        "name": "westend",
        "playlist": "westend.m3u8"
      },
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              -123.14892768859862,
              49.28925041973686
            ],
            [
              -123.1377696990967,
              49.2766528015807
            ],
            [
              -123.11536788940428,
              49.29014612779744
            ],
            [
              -123.13339233398438,
              49.296527576688376
            ],
            [
              -123.14892768859862,
              49.28925041973686
            ]
          ]
        ]
      }
    }
  ]
}

```

## Controlling Playback using Motion Detection

With devices that have a camera, Video Kiosk can use the camera to detect motion and start playing. It will stop playing media if no motion is detected for the specified timeout.

### Requirements:

- To use this feature, your Android device must have a front-facing camera.
  - Most phones and tablets come with a camera installed. Most Android TVs do not come with a camera installed, but you can buy a third-party USB camera to connect to the Android TV.

To use motion detection playback,

1. Enable Video Kiosk motion detection. Go to **Video Kiosk Settings > Schedule > Detect motion** and enable “Use the camera to detect motion”
2. Optionally, configure the timeout when no motion is detected. (Default is 5 minutes.)

For information on Settings, refer to [Configuring the Settings]

## Monitoring Battery State

Some devices, notably tablets, have batteries. You can configure Video Kiosk to play content based on the percentage battery charge remaining and whether or not the battery is charging (i.e. the device is plugged in to a charger).

1. Configure Video Kiosk battery monitoring. Go to **Video Kiosk Settings > Schedule > Monitor battery charge** and configure the behavior when the battery is not charging and when the battery charge is low. For information on Settings, refer to [Configuring the Settings]

## Changing the Playback Mode

Usually the Playback Mode is set to “Play next”, and Video Kiosk plays the next media item when the current media item finishes. However, you can change this behaviour to play the previous, to pause after play or to loop play and require a swipe event to continue.

To change the playback mode,

1. Go to **Video Kiosk Settings > Interactive Playback > Mode**

and set the Playback Mode.

For information on Settings, refer to Configuring the Settings section of this document.

# Controlling the Display





Display features control how your content looks on the device. You can add an overlay over your image (like your company logo in the corner) or add a background or background music. You can also customize the time images are displayed, configure timers to fade between media files and change the screen orientation.

For information on Settings, refer to Configuring the Settings section of this document.

## View Options - Screen Orientation, Timers and Transitions

### Screen orientation

Video Kiosk allows you to set the screen orientation for just the media playback screen. The other views, such as settings, are not affected. (If you want to change all the view orientations, you can use the device System Settings). The supported orientations are:

<b>Default</b>	The default screen orientation set by the manufacturer of your device	
<b>Landscape</b>	Screen width is wider than is screen height	
<b>Portrait</b>	Screen height is taller than screen width	
<b>Reverse Landscape</b>	Landscape orientation with device turned 180 degrees from landscape	
<b>Reverse Portrait</b>	Portrait orientation with device turned 180 degrees from portrait	

To change the screen orientation, go to **Video Kiosk Settings > View Options > Screen Orientation**.

### Image Display Options

#### Image Display Time

If your media folder includes still images, each one will be displayed, by default, for 30 seconds.

- To increase or decrease the display time, go to **Video Kiosk Settings > Timers > Image display time** and change the display time (in seconds).

- If you set the time to 0, Video Kiosk will display an image and then pause until the screen is touched. You can use this feature to pause the media loop and wait for user interaction prior to continuing.
- To override the default display time for individual images, refer to the "Controlling Display using file name" section of this document. As an example, you might want to do this if there is a lot of text on one slide and you want to give viewers more time to read it.)

## Image Fade Time

**Note:** Requires Android v4.2 or later

If your media folder includes still images, each one can fade in and fade out over some number of milliseconds. To increase or decrease this fade time, go to **Video Kiosk Settings > Timers > Image fade time** and change the display time (in seconds).

## Other Image Display Options

Video Kiosk provides settings for handling images, such as:

- Image View Scale Type – supports many scaling options, such as:
  - Pan and Zoom (Ken Burns) effect,
  - Fit XY, Fit Start, Fit Center, Fit End
  - Center, Center Crop, Center Inside
- Image resizing – scales images to fit rotated screen
- JPG dithering – use dithering when decoding
- Use the EXIF Orientation META data to control image rotation – Rotate images based on EXIF Orientation in file meta data
- Controlling max megapixels – reduces quality of images larger than your max limit

To change the image settings, go to **Video Kiosk Settings > View Options**.

## Video Display Options

### Video Fade Time

**Note:** Requires Android v4.2 or later

If your media folder includes video files, each video can fade in and fade out over some number of milliseconds. To increase or decrease this fade time, go to **Video Kiosk Settings > Timers > Video fade time** and change the display time (in seconds).

*NOTE: Fading video is resource intensive. On low end Android sticks enabling Video fading can result in choppy video playback. If you experience choppy video playback, set the Video fade time back to 0.*

### Other Video Display Options

Video Kiosk provides optional 4K Display support, which enables the 4K display support provided by the particular device. For more information, refer to

**Settings > View Options > Request Best Resolution** in the Settings section of this document.

## Using textured video views and / or pause playback to reduce black flash

*Please note that some MP4 video encodings do not work with textured video views and you may get a blank or green screen during playback. If this happens, you need to turn off textured video views.*

The chip set in some devices, typically Amlogic-based, display a black flash before and after playing each video. This is because the 3D rendering pipeline is used to display standard video views. As an alternative, you can use textured video views, which use the 2D rendering pipeline, to reduce or eliminate this black flash. Please note that using the 2D rendering pipeline is slower so on some low-end devices enabling this setting can result in choppy playback. On Android v4.2 and later, you can enable this setting to reduce or eliminate the black flash. To do this, go to **Video Kiosk Settings > View Options > Use textured video views** and check the checkbox.

In addition, when Video Kiosk reaches the end of a video, it invokes the method `StopPlayback()` which can also result in a black flash on some Amlogic based devices. You can configure Video Kiosk to use `Pause()` instead of `StopPlayback()` which eliminates this second source of black flashes on some devices. To do this go to **Video Kiosk Settings > View Options > Pause playback** and check the checkbox. Unfortunately, on some devices, enabling his setting can result in playback errors so be careful and only enable this setting if you are still getting black flashes after enabling textured video views.

## Web Display Options

### Web Fade Time

**Note:** Requires Android v4.2 or later

If your media folder includes still .url files, each web page can fade in and fade out over some number of milliseconds. To increase or decrease this fade time, go to **Video Kiosk Settings > Timers > Web fade time** and change the display time (in seconds).

### Other URL Display Options

Video Kiosk provides settings for handling media, such as:

- Preload URLs – Looks ahead and preloads URL files in the background

To change the media handling settings, go to **Video Kiosk Settings > View Options**.

## Overlays

An overlay is a transparent slide that overlays something over the screen. It has two main uses:

- An Overlay is often used to overlay the screen with a small digital graphic (aka a digital on-screen graphic, digitally origination graphic (DOG), bug, network bug), usually a logo in the corner of the screen. This used for brand identification. It looks something like:



- An Overlay with the addition of hotspots is used to turn your device into an interactive kiosk. For example, you could make a hotspot file for the image above so that when you touched the product logo in the bottom corner of the slide, you would be taken to the Video Kiosk page on the Burningthumb website. For more information about creating an Overlay with Hot Spots, refer to the “Touch Screen Interactive Display Mode” chapter in this document.

## How to use an overlay file

To use an Overlay, follow these steps:

1. **Create the file.** Create a PNG file with the same frame size as your display. For example, if you are using a 720p display, you would create a PNG file that is 1280×720 in size. Somewhere in the PNG file put your logo, or whatever image you want to use to overlay the video. End the file name in “.overlay.png” so that Video Kiosk recognizes it as an overlay and not as an image to display as part of the content loop.

For information on how to create a file with a transparent background, refer to the guides at:

<https://burningthumb.com/how-to-guides/>

2. **Move the file.** Move the file onto the device or synchronize it using a cloud service. (Standard Folder Layout – move to “videokiosk/overlay/”)
3. **Configure the Path.** If you are using a Custom Folder Layout, configure Video Kiosk with the location of the overlay file. (If you are using the Standard Folder Layout or One Folder Layout, this step is not required.)  
Go to **Video Kiosk Settings > Overlay > Path to Overlay folder** and set the path to the overlay folder.
4. **Select the File.**  
Go to **Video Kiosk Settings > Overlay > Overlay file** and select the overlay to use.
5. Optionally, if the overlay image is a different size than the screen dimensions you may want to adjust the scale type using **Video Kiosk Settings > Overlay > Overlay image scale type**

Video Kiosk will overlay your media with the content of the overlay file.

There is a sample overlay file included in the **Video Kiosk Media Sample Pack**. You can use it to see what an overlay might look like in your project before creating one of your own.

Download it from:

<https://burningthumb.com/wp-content/uploads/video-kiosk-media-sample-pack.zip>

## Background Images and Audio

Background features control the image or audio behind your content.

### Background Images

Video Kiosk supports background images that will be displayed behind your media as it is playing and behind digital signage. You can use no background image (the default), one background image, or cycle between a number of background images.

To use a single background image, follow these steps:

1. **Create the File.** Create a background image file in png or jpg format. We recommend you make it the same size as your screen (e.g. 1280×720 in size)  
or  
Use one from the **Video Kiosk Media Sample Pack**. Download it from:  
<https://burningthumb.com/wp-content/uploads/video-kiosk-media-sample-pack.zip>
2. **Move the File.** Move the file onto the device or synchronize it using a cloud service.  
(Standard Folder Layout – move to “videokiosk/background/”)
3. **Configure the Path.** If you are using a Custom Folder Layout, configure Video Kiosk with the location of the background file. (If you are using the Standard Folder Layout or One Folder Layout, this step is not required.)  
Go to **Video Kiosk Settings > Background > Path to Background folder** and set the path to the background folder.
4. **Select the File.**  
Go to **Video Kiosk Settings > Background > Background file** and select the background file to use.

To cycle through a number of background files, follow these steps:

1. Using the instructions above, create and move the background files onto the device and configure the device.
2. Go to **Video Kiosk Settings > Background > Next Background Time** and set the number of seconds between backgrounds.

### Example: Sample Configuration

- Put 3 background files (background-a.png, background-b.png and background-c.png) in the background folder
- Set the Next Background Time to 20 seconds

Video Kiosk will cycle through each of the three background images, displaying each for 20 seconds.

## Background Audio

Video Kiosk supports background audio. You can play streaming audio or a local audio file that Video Kiosk will play in a loop. By default, Video Kiosk will look for a local audio file named `backgroundaudioloop.mp3` in your Background folder and if it is found it will be played automatically with no additional configuration required.

To use background audio, follow these steps:

1. **Create the File.** Create a background audio file and name it “backgroundaudioloop.mp3”
2. **Move the File.** Move the file onto the device or synchronize it using a cloud service. (Standard Folder Layout – move to “videokiosk/background/”)
3. **Configure the Path.** If you are using a Custom Folder Layout, configure Video Kiosk with the location of the background file. (If you are using the Standard Folder Layout or One Folder Layout, this step is not required.)  
Go to **Video Kiosk Settings > Background > Path to Background folder** and set the path to the background folder.
4. To enable background audio, Check the **Enable background audio** checkbox
5. Click the **Done button**.

To disable background audio playback:

1. Go to **Video Kiosk Settings > Background > Audio URL**
2. To disable background audio, Uncheck the **Enable background audio** checkbox
3. Click the **Done button**.

## Controlling Display using file name meta data

One of the ways you can control Video Kiosk is by adding meta data to the file name of your media file. This feature is used when you want to do one of the following:

- Customize the Image Display time for some of your images.
- Display Titles as an overlay for your media files
- When in split-screen mode, display some of your images or video configured to play in Split-Screen Area A in full screen instead

## Customizing the Image Display Time for one image file

You might want to use this feature if you were showing images and some needed more time than others. For example, if you had some photos, with a slide in the front explaining where they were

taken, you might want to use a longer display time for the explanation slide to give your viewers time to read the text.

To change the image display time on an image from the default of 30 seconds to 45 seconds, add **".timeout=45."** to the filename.

For example, change the filename of  
vacationontheislandtext.jpg  
to  
vacationontheislandtext.timeout=45.jpg

That one image will now be displayed for 45 seconds while the rest of the images will be displayed for 30 seconds. For exact syntax, refer to the "File Name Meta Data Syntax Guide" below.

## Displaying Titles in the corner of each media file

To display Titles in the bottom right of each file, do the following:

1. Go to **Video Kiosk Settings > View Options > Show Media name** and enable this feature.
2. For each slide, add Title meta data to the filename. For exact syntax, refer to the "File Name Meta Data Syntax Guide" below.

Please note that if you don't specify Title meta data for a media file, the filename will be displayed in the title banner. This can be useful when testing your content loop.

## Switching between Split Screen and Full Screen when playing a media file

When you are using Split Screen mode, you may want to display some media items in full screen instead of in Split Screen Area A, the default when using split screen mode. To display a media file in full screen mode, **".fullscreen=true."** to the filename. When the next media item is played, it will revert to playing in Split Screen Area A.

## File Name Meta Data Syntax Guide

This section summarizes the meta data that you can embed in the file name of a media item where each piece of meta data uses the form:

**.key=value.**

Note that the key / value pair is surrounded by dots. This is a required element of the syntax. If the dots are not present, Video Kiosk will ignore the key / value pair.

Key	Value Type	Description
timeout	numeric	Set the timeout for images. This overrides the default Image Time specified in the settings.

Key	Value Type	Description
		(Valid for images only. Video and Web media items have different mechanisms for setting their time values.)
title	text	Set the string that will be displayed if the <b>View Option &gt; Show media name</b> setting is enabled. <b>Note:</b> If a title is not specified as meta data in the file name and the Show media name option is enabled, the file name will be displayed as the media name. This will include any meta data specified for this file. (Valid for images, video and web media.)
fullscreen	boolean	Set the media item to be displayed in full screen when Widget Mode has been enabled. If Widget Mode has not been enabled, this has no effect. If Widget Mode has been enabled, this media item will be displayed in full screen mode and when it is finished playing, the display will return to displaying media in Widget Area A. Any value, other than <b>true</b> , is treated as <b>false</b> . (Valid for images, video and web media.)

## Examples:

Here's an example where we have three files, two containing meta data.

```
palmtrees.jpg
thebeach.title=Waikiki Beach.timeout=60.fullscreen=true.jpg
therain.fullscreen=true.jpg
```

The user has configured Video Kiosk with

- an "Image Display Time" of 30 seconds,
- has enabled the "Show media name" option,
- has enabled Split Screen mode with the images being displayed in Widget Area A.

The results would be the following:



The first image - **palmtrees.jpg**

- will display for 30 seconds
- will display the title "palmtrees"
- will display in Widget Area A



The second image -  
**thebeach.title=Waikiki Beach.timeout=60.fullscreen=true.jpg**  
- will be displayed for 60 seconds, overriding the default timeout  
- will display the title "Waikiki Beach"  
- will be displayed in full screen instead of just Widget Area A



The third image -  
**therain.fullscreen=true.jpg**  
- will be displayed for 30 seconds  
- will display the title "therain" name will be displayed as "therain.fullscreen=true", since Title has not been set  
- will be displayed in full screen instead of just Widget Area A

## Legacy Support Display Features

Video Kiosk includes some features to work around issues with using old video or old Android devices.

### Stretch video

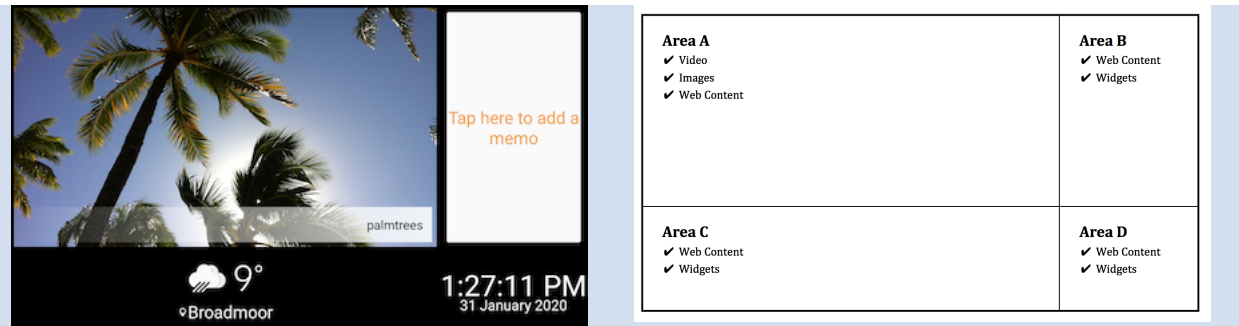
Old video is narrower (960x720) than the new standard video size (1280x720), which causes black bands to be displayed in the vacant space on the sides. You can use the Stretch Video feature to eliminate the black bands. To do this, go to **Video Kiosk Settings > Legacy > Stretch video** and check the checkbox.

### First frame render time

This controls how long (1/100th seconds) the first video frame takes to render.

Older Android devices can flicker as they attempt to render the first frame of video. This flickering can be eliminated by increasing the first frame render time. To do this, go to **Video Kiosk Settings > Legacy > First frame render time** and increase the render time.

# How to Convert to Split Screen Display Mode



Video Kiosk has a Split Screen Mode, which divides the playback screen into 4 areas, A, B, C and D, allowing you to play your content loop in area A, while you run widgets or URL-based content in areas B, C, and D. Video Kiosk includes some built-in widgets you can use.

## How to use Split Screen Mode

1. Enable Widget Mode.
2. Configure the widget Areas B, C and D with which widgets | URLs to display.
3. Optionally, adjust the widget area layout.

## Configuring the Split Screen Display

To configure Split Screen mode,

1. **Enable Widget Mode.**

If you haven't already done so, enable Widget Mode.

Go to **Video Kiosk Settings > Widgets > Enable widgets** and check the checkbox.

Video Kiosk will re-launch and display a 4-area split screen (Referred to as Areas A, B, C and D.)

2. **Configure the widget areas.**

Configure the widget Areas B, C and D with the widgets | URLs to display. (You can use the built-in widgets, third-party widgets or URLs.)

For each of Areas B, C and D, go to **Video Kiosk Settings > Widgets > Choose Area x Widget** and select and configure the widget(s) and URL(s) to display.

Each widget area can display multiple app widgets or URL-based content.

**Built-In Widget.** To Display a Built-In widget, click the "BUILTIN" button. Choose a widget (or Widgets) from the list. To configure each widget, refer to the "Using the Built-in Video Kiosk Widgets" section below.

Each widget area can display multiple app widgets or web-based content. Add any other widgets to be displayed in that Area.

**Default Min(utes).** If you've configured more than one widget in an area, each will be displayed for this amount of time, in minutes. The default is 60 minutes. Modify, if required.

**Widget.** To Display a third-party widget, click the "WIDGET" button. Choose a widget (or Widgets) from the list. To configure each widget, refer to the documentation that came with your widget.

Each widget area can display multiple app widgets or web-based content. Add any other widgets to be displayed in that Area.

**Default Min(utes).** If you've configured more than one widget in an area, each will be displayed for this amount of time, in minutes. The default is 60 minutes. Modify, if required.

**URL.** To Display a URL-based content, click the "URL" button. The web URL can be any valid protocol including http://, https:// (remote content) and file:// (local content). Enter each URL.

**Default Min(utes).** If you've configured more than one URL in an area, each will be displayed for this amount of time, in minutes. The default is 60 minutes. Modify, if required

3. Optionally, configure the "**Monitor Internet**" feature.

If your widgets | URLs are dependent on the Internet, they may behave in an unexpected manner if Internet connection is lost. You can configure Video Kiosk to monitor the Internet and return to full screen mode when Internet connection is lost. To enable this feature, go to **Video Kiosk Settings > Widgets > Monitor Internet**, and tick the box.

4. Optionally, configure the "**Arrange in rows**" option.

The default is to arrange your widgets in columns.

5. Optionally, **adjust the widget area layout and the widget area padding.**

For more information, refer to the "Customizing Widget Area Layout" section below.

## To Return to Full Screen Mode

To return to Full Screen Mode,

1. Go to **Video Kiosk Settings > Widgets > Enable** and uncheck the checkbox.

Video Kiosk will return to Full Screen mode.

## Default Display

Each area has a default it will display. When split screen mode is enabled, the following will be displayed:

- Area A will display your content loop
- Areas B, C and D will display your background, or a black screen if no background is configured

## Displaying Widgets

### Widget Considerations - Where to find widgets

- Video Kiosk provides built-in widgets for RSS, Date and Time, Image folder, and URL
- Widgets are available from the Google Play Store.
- You can develop your own widgets and side load them onto your Android devices.

### Widget Considerations - Widget Readability

Because Areas B, C and D are of different sizes, you need to give some consideration to how well the widget will be seen when placing the widget in one of the areas. When deciding on which area to place a widget, consider how the widget was intended to be displayed (landscape, portrait or square), the device's display size (TV or tablet or phone) and how far away the viewer will be from the screen.

### Using the Built-in Video Kiosk Widgets

Video Kiosk provides a set of built-in widgets. These include Date and Time, Image Folder, URL, and RSS widgets.

When you pick a built-in widget, the settings for that widget are displayed.

You can configure the widget using these settings or using an XML file. For more information about using an XML file, refer to Appendix D – Built-in Widget XML File Settings.

### *Date and Time Widget*



Use the Date and Time widget to display the Date and / or the Time with or without background images.

When you pick the widget, the settings will be displayed.

### *Date and Time Widget Settings*

*Set if you're configuring the widget using an XML File*

*XML Settings* – path to a file containing XML settings for the widget

*Set if you're configuring the widget manually*

*Image folder* – pick the folder that contains the background images

*Image display time* – the time, in seconds, to display each image

*Time font size* – the font size of the time

*Time switch* – turn on or off the display of the time

*Date font size* – the font size of the date

*Date switch* – turn on or off the display of the date

*Font color* – the font color

When you select a widget in any of the widget areas, the settings will be displayed. To Configure the Date and Time Widget:

1. (Optional) Make your background image file(s) and put in a folder called "Images". Move the folder onto the device or synchronize it using a cloud service. (Standard Folder Layout – move to "videokiosk/widget/date/")
2. Configure the Widget Settings

*If you're using an XML setting file to configure the widget*

*XML Settings* – path to a file containing XML settings for the widget (Not required if you're configuring the widget manually)

*If you're configuring the widget manually*

*Image folder* – pick the folder that contains the background images or use the default  
*Image display time* – the time, in seconds, to display each image, or use the default (30 seconds)

*Time font size* – the font size of the time (Default is 32 dpi.)

*Date font size* – the font size of the date (Default is 16 dpi.)

*Font color* – the font color. In ARGB. The default is Black (#FFFFFFFF).

3. Press "OK"

### **Image Folder Widget**



Use the Image Folder widget to display a group of images from a folder in rotation. The widget will scale the images to fit Areas B, C and D. When you pick the widget, the settings will be displayed.

#### **Image Widget Settings**

*Set if you're configuring the widget using an XML File*

*XML Settings* – path to a file containing XML settings for the widget

*Set if you're configuring the widget manually*

*Image folder* – pick the folder that contains the background images

*Image display time* – the time, in seconds, to display each image

When you select a widget in any of the widget areas, the settings will be displayed. To Configure the Image Widget:

1. Make your image file(s) and put in a folder called “Image”. Move the folder onto the device or synchronize it using a cloud service. (Standard Folder Layout – move to “videokiosk/widget/”)
2. Configure the Widget Settings

*If you’re using an XML setting file to configure the widget*

*XML Settings* – path to a file containing XML settings for the widget (Not required if you’re configuring the widget manually)

*If you’re configuring the widget manually*

*Image folder* – pick the folder that contains the images or use the default

*Image display time* – the time, in seconds, to display each image, or use the default

If using the XML Settings to configure the widget (See Appendix D)

3. Press “OK”

## RSS Widget



Use the RSS widget to display a list of RSS News feeds from remote servers. When you pick the widget, the settings will be displayed.

### RSS Widget Settings

*Set if you’re configuring the widget using an XML File*

*XML Settings* – path to a file containing XML settings for the widget

*Set if you’re configuring the widget manually*

*Kind* – one of Tall | Wide | Marquee

*Next download hours* – the number of hours to wait between loading the feed from the server

*Next URL minutes* – the number of minutes to wait between displaying feeds

*Max document items* – the maximum number of news items to display per feed

*Next item seconds* – the number of seconds to wait between displaying document items

*Title font size* – the font size of the item title

*Description font size* – the font size of the item description

*Font color* – the font color

*Hide when value is* – if the title or description contain this value, they will be hidden

*Justify title* – one of Left | Center | Right and on Android v8 or later Full

*Justify description* – one of Left | Center | Right and on Android v8 or later Full

*Background* – a tint color (ARGB) and path to 1 or 2 background images (they rotate every

hour)

*Feeds* – the list of RSS Feed URLs

When you select a widget in any of the widget areas, the settings will be displayed. To Configure the RSS Widget:

1. (Optional) Make your background image file(s) and put in a folder called “Images”. Move the folder onto the device or synchronize it using a cloud service. (Standard Folder Layout – move to “videokiosk/widget/rss/”)
2. Click the “Settings” tab and configure the Widget Settings

*If you’re using an XML setting file to configure the widget*

*XML Settings* – path to a file containing XML settings for the widget (Not required if you’re configuring the widget manually)

*If you’re configuring the widget manually*

*Kind* – one of Tall | Wide | Marquee (default is wide)

*Next download hours* – the number of hours to wait between loading the feed from the server (default 1 Hour)

*Next URL minutes* – the number of minutes to wait between displaying feeds (default 30 mins)

*Max document items* – the maximum number of news items to display per feed (default 50)

*Next item seconds* – the number of seconds to wait between displaying document items (default 300 seconds)

*Title font size* – the font size of the item title (default 26 px)

*Description font size* – the font size of the item description (default 16 px)

*Font color* – the font color (default #FFAAAAAA)

*Hide when value is* – if the title or description contain this value, they will be hidden

*Justify title* – one of Left | Center | Right | Full

*Justify description* – one of Left | Center | Right | Full

*Background* – a tint color (ARGB) or a path to 1 or 2 background images (they rotate every hour)

3. Click the “Feeds” tab and configure the Feeds List
4. Press “OK”

## URL Widget



Use the URL widget to display a local file:// or remote http:// HTML / CSS/ Javascript based widget. When you pick the widget, the settings will be displayed.

### URL Widget Settings

*Set if you're configuring the widget using an XML File*

*XML Settings* – path to a file containing XML settings for the widget

*Set if you're configuring the widget manually*

*URL* – the local file:// or remote http:// URL to display

When you select a widget in any of the widget areas, the settings will be displayed. To Configure the RSS Widget:

1. Configure the Widget Settings

*If you're using an XML setting file to configure the widget*

*XML Settings* – path to a file containing XML settings for the widget (Not required if you're configuring the widget manually)

*If you're configuring the widget manually*

*URL* – the local file:// or remote http:// URL to display

2. Press "OK"

## Displaying URL-based Content

Because Areas B, C and D are of different sizes, you need to give some consideration to how well the URL-based will be seen when you link to it in one of these areas. When deciding on which area to use, consider how the content is intended to be displayed (landscape, portrait or square), the device's display size (TV or tablet or phone) and how far away the viewer will be from the screen.

## Customizing Widget Area Layout

You can use the Video Kiosk default layout, or you can customize the layout by adding or removing padding between the areas or by changing the size of the widget areas.

### Adjusting Widget Area Size

Widget Area Size is the boundaries of the four Areas (Areas A, B, C and D) including the area used by the widget and the padding.

By default Video Kiosk assumes a 1080p display in landscape orientation with a screen resolution of 1920x1080. The horizontal weights are 1426 and 494 and the vertical weights are 800 and 280.

So on a 1080p display the pixel sizes will be as follows Area A: 1426x800, Area B: 494x800, Area C: 1426x280 and Area D: 494x280. The best way to understand | use the layout weights is to determine the horizontal width and vertical height of your display based on its orientation and then set the weights to the number of pixels that you want in your layout.

To adjust a widget's area size,

1. Go to **Video Kiosk Settings > Widgets > Horizontal layout** and adjust the horizontal weight for the widget's area.
2. Go to **Video Kiosk Settings > Widgets > Vertical layout** and adjust the vertical weight for the widget's area.

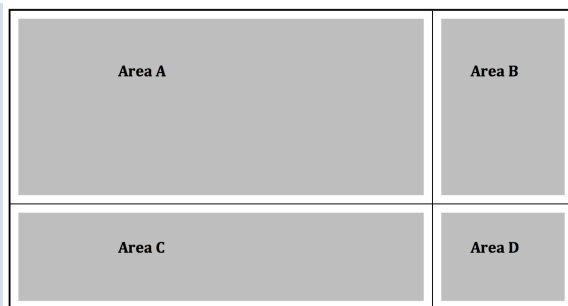
### Adjusting Widget Area Padding

Padding is the area between the Widget area boundaries and the sides of the Widget. Each widget area has a default padding amount which allows the background to show through between the widget areas and aligns the widget within the area. If you'd like to change a widget's placement slightly within it's area, use the padding feature. Please note that the padding can be both positive and negative. For web URL content, additional padding and formatting may need to be included in the html description of the content.

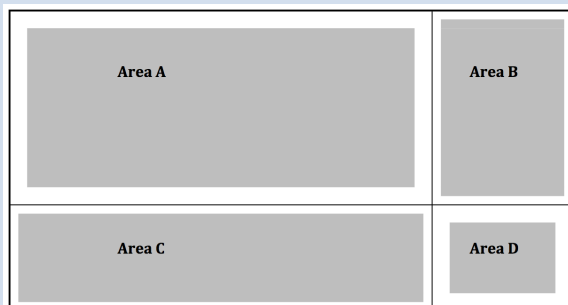
To adjust a widget's padding,

1. Go to **Video Kiosk Settings > Widgets > Area X Padding** and adjust the padding in the widget's area.

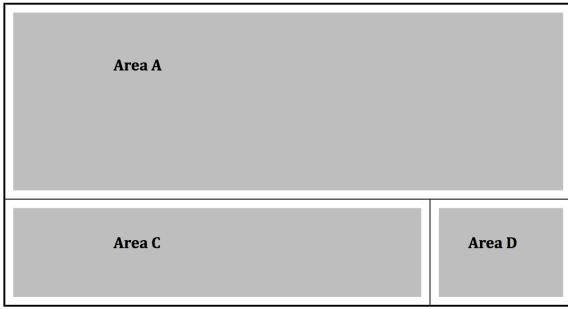
### Example: Customizing Screen Widget Areas



In this is an example, the default widget area layout and default widget padding has been used.



In this example, the default widget area layout is used and the padding for Area A and Area D have been increased.



In this example, the default widget padding has been used. To allow Area A to use the entire width of the screen, the horizontal weight of Area B has been set to 0, while the horizontal weight of Area A has been set to the maximum for the screen resolution.

# How to Convert to Touch Screen Interactive Kiosk Display Mode

Video Kiosk's playback screen supports three different modes in which to display media:

- Full Screen Mode, as well as two more advanced modes
- Split Screen Mode
- Touch Screen Interactive Kiosk display mode.

Video Kiosk supports Interactive Kiosk Mode, which turns your device into an interactive Touch Screen or mouse-controlled kiosk, displaying web URLs or Apps.

## How to turn your device into an interactive kiosk

1. Create an overlay. For more information, refer to [Overlays, p. ]
2. Create a Hotspot File using the instructions below.
3. Once you've created the Hotspot File, test the hotspots to ensure they are behaving as expected. For more information, see the instructions below.
4. Lock down the interactive kiosk to prevent the user from overriding the content you wish to display. For more information, refer to [Enhanced Access Management Set Up, p. ]

## Understanding Interactive Hotspots

Video Kiosk supports using a hotspot xml file, with hotspots and/or key codes defined, to allow the user to interact with the device. Interactions can also be tracked.

A hotspot is a rectangle somewhere on the screen or a key code and its associated URL | App that is used to control interaction behavior. When the hotspot is activated, the URL will be displayed using the integrated kiosk web view or the App will be launched as a new activity on top of Video Kiosk. (The URL can point to a file on the device or a file on the Internet).

- ***Using a Hotspot to open a Web Page***

When the hotspot associated with a web URL is activated, the web page will be displayed in an integrated web view that limits the users web browsing to the URL you specified in the hotspots. To ensure your kiosk is showing your content loop, Video Kiosk monitors how long the device has been displaying the hotspot web content and returns to showing your content loop after a configurable amount of time. This ensures that an interactive device abandoned by a user while displaying web content returns to showing your content loop. You can change this inactivity timeout in the Video Kiosk settings.

- ***Using a Hotspot to open an App***

When a hotspot associated with an App package name is activated, the App will launch and Video Kiosk will lose control of the device until the back button or home button is used to return to Video Kiosk.

*If you plan to use a hotspot associated with an App package name, there is no automatic mechanism that ensures a device abandoned by a user will return to the loop and the user must take specific action to return to Video Kiosk so that the loop continues to play.*

- **Using Hotspots with Split Screen Mode**

If you are using Video Kiosk in split screen mode, you can specify the URL | App for each App Widget. Video Kiosk will first look at App Widgets configuration, and use any values found there while falling back to the XML file configuration if nothing was specified for the App Widget.

## Activating Hotspots

Hotspots are activated either by a screen touch or by pressing the key associated with a pre-assigned keycode on the keyboard or remote control. The area of the screen to touch or the key to press is defined in the Hotspot.xml file.

A key code is the Android key code that corresponds with an individual key on an input device. When the key on the device that corresponds to the Android key code is touched, the URL will be displayed using the integrated kiosk web view or the App will be launched as a new activity on top of Video Kiosk. For more information, refer to the "Controlling the Device" section of this document.

## How to Define Hotspots using an overlay xml Hotspot file

Define each hotspot and its associated web URL or App package name using an overlay file (which displays the hotspots to the user) and an overlay hotspot xml file (which tells Video Kiosk what to do when a particular hotspot is activated).

To use hotspots, follow these steps:

1. If you haven't already done so, create an .overlay.png file. (see above)
2. In a Text Editor, create an XML Hotspot file and use it to control interaction behavior. When you have finished editing the file, change the file name to the same name as your overlay file and change the extension to ".xml" so Video Kiosk will recognize it as the hotspot file that goes with the overlay file.  
For example, if your overlay file is called "videokiosk.overlay.png", name your XML hotspot overlay file, "videokiosk.overlay.png.xml".  
For more information about the xml file syntax, refer to the **Video Kiosk XML Hotspot File Syntax Guide** below
3. Move the XML hotspot file onto the device. Put the XML hotspot file in the same folder as the Overlay file so that Video Kiosk will find it automatically.
4. Optionally, configure the timeout for the WebView in Video Kiosk. If a user of an interactive device leaves while the device is still displaying web content, Video Kiosk will return to your content after 30 seconds. To increase or decrease this wait time, go to **Video Kiosk Settings > Overlay > Web Activity timeout** and change the timeout (in seconds). For more detailed information about using this setting, refer to [Configuring the Settings].

5. Optionally, configure the Hide overlay option, which controls what will happen to the overlay if the screen is not touched for more than 15 second. Set this, go to Video Kiosk Settings > Overlay > Hide overly. For more detailed information about using this setting, refer to [Configuring the Settings ].
6. Optionally, disable WebView navigation.  
Note: If you disable WebView navigation, you are responsible for building navigation into your web page(s), including invoking the method `window.VideoKiosk.closeWebView()`; to return to Video Kiosk’s media loop playback. For more detailed information about using this setting, refer to [Configuring the Settings ].
7. If you are using Video Kiosk in Split Screen mode and have configured the XML file to use the `<widgetarea></widgetarea>` tag, then configure the URL | App and interaction key for each App Widget that you add to each Widget Area. If you don’t configure those items in the Widget Area configuration, Video Kiosk will use the configuration from the XML file.

*Note: If you are using Widget Areas to define the URL | App you should align the hotspots so that they appear over the widget areas. This is not required, but it probably makes the most sense to do it this way.*

## Video Kiosk XML Hotspot file Syntax Guide

<code>&lt;hotspots&gt;</code> <code>&lt;/hotspots&gt;</code>				A list of hotspots
	<code>&lt;hotspot&gt;</code> <code>&lt;/hotspot&gt;</code>			A single hotspot. To be active, each hotspot requires a destination url or app is specified and that an activating rectangle and/or key code is specified.
		<code>&lt;url&gt;</code> <code>&lt;/url&gt;</code>		Destination web URL used for a single hotspot
		<code>&lt;app&gt;</code> <code>&lt;/app&gt;</code>		Destination App bundle identifier used for a single hotspot.  The special value <b>builtin.wifi</b> will open a built-in WIFI configuration Activity.
		<code>&lt;command&gt;</code> <code>&lt;/command&gt;</code>		A hotspot command can be one of the following: pause – playback will pause resume – playback will resume nextfile – go to the next file previousfile – go to the previous file nextfolder – go to the next folder previousfolder – go to the previous folder
		<code>&lt;depth&gt;</code> <code>&lt;/depth&gt;</code>		When using the commands nextfolder and previousfolder commands, this integer value limits how many folders deep to look. This allows you to control whether sub-folders are considered next or previous folders. Valid Values: 0 - next is next folder at same folder level 1 - Next folder is next sub-folder inside

				the folder 2 - Next folder is next sub-folder inside the sub-folder
		<widgetarea> </widgetarea>		Look at the specified Widget Area (B   C   D) for the url, app, and interaction key. The values will be selected based on the current widget being displayed in the Widget Area. If values are found in the Widget configuration, use them. Otherwise use the ones specified in this file.
		<keycode> </keycode>		The Android key code, used to activate a hotspot using a keyboard
		<rect> </rect>		The screen rectangle for the hotspot defined using numbers between 0 and 1 as a fraction of the screen
			<top> </top>	Rectangle top
			<left> </left>	Rectangle left
			<bottom> </bottom>	Rectangle bottom
			<right> </right>	Rectangle right
		<interactionkey> </interactionkey>		(Optional) A string that identifies the hotspot. If this tag is set, the interactions for the hotspot will be counted and reported. If this tag is not set, the interactions for the hotspot will not be counted.
		<password> </password>		(Optional) A SHA 512 encoding of the password that must be entered to access this APP or URL
		<pinned> </pinned>		(Optional) A boolean that determines if Video Kiosk tries to keep the launched App pinned. The default value is false. On Android 9 setting the value to true has no effect because Android v9 requires a new task which will result in the display unpinning.

## Hotspot .xml file example

For more information about overlays and XML Hotspot files or to download a sample kit, refer to the tutorials at:

<http://burningthumb.com/apps/video-kiosk/tutorials/>

Here is an example of what a file that contains 2 hotspots could look like.

- The first hotspot's destination is the URL "http://burningthumb.com". It can be triggered by touching the hotspot rectangle or by pressing the key corresponding to key code 51. The interactions to this hotspot are counted and reported in the Management Report with the identifying string of "burningthumb".

- The second hotspot's destination is the App "Calculator2". It can be triggered by touching the hotspot rectangle or by pressing the key corresponding to key code 29. The interactions to this hotspot are neither counted nor reported in the Management Report.

```
<?xml version="1.0" encoding="UTF-8"?>
<hotspots>
  <hotspot>
    <url>http://burningthumb.com</url>
    <rect>
      <top>0.55</top>
      <left>0.05</left>
      <bottom>0.95</bottom>
      <right>0.45</right>
    </rect>
    <keycode>51</keycode>
    <interactionkey>burningthumb</interactionkey>
  </hotspot>
  <hotspot>
    <app>com.android.calculator2</app>
    <rect>
      <top>0.55</top>
      <left>0.55</left>
      <bottom>0.95</bottom>
      <right>0.95</right>
    </rect>
    <keycode>29</keycode>
  </hotspot>
</hotspots>
```

## Hotspot Interaction Tracking

Video kiosk can track interactions with hotspots. If you want Video Kiosk to track interactions, specify an interaction key in the Hotspot.xml file. To report on interactions, enable interaction reporting on the Video Kiosk management interface.

If you are using Video Kiosk in split screen mode, you can specify the interaction key for each App Widget. Video Kiosk will first look at App Widgets configuration, and use any values found there while falling back to the XML file configuration if nothing was specified for the App Widget.

## Hotspot .xml file Verification help

To help test and debug your .xml file, Video Kiosk has an optional setting you can use to highlight your hotspots on your screen so that you can ensure they are in the correct place and functioning as expected. Remember to turn this feature off before releasing the device to users.

To turn verification mode on

1. Go to Video Kiosk Settings > Overlay > Highlight hotspots and check the checkbox.

To turn verification mode off

1. Go to Video Kiosk Settings > Overlay > Highlight hotspots and uncheck the checkbox

# Controlling the Device

For information on Settings, refer to [Configuring the Settings]

## TV Control using HDMI CEC (Android v5 and later)

In order to play content, the TV must be turned on and the correct video input must be selected.

- *On any Android 4+ device*, Video Kiosk will automatically start playing content when scheduled (if no calendar or schedule is being used then content will always play). When the scheduled event has ended, Video Kiosk will stop playing content, dim the screen, and optionally display a clock.
- On Android devices running Android 5+, Video Kiosk can turn the TV on
- *On Android devices running Android 5+*, Video Kiosk can control the selected HDMI input.
- *On Android devices running Android 6+*, Video Kiosk can turn the TV on, select the correct input, and turn the TV off instead of dimming the screen.

### Using HDMI-CEC to control the TV

Video Kiosk can send HDMI-CEC commands to keep the TV on and the Android TV device HDMI input selected when media is being displayed (requires at Android v5.1+) and/or turn the TV off (put the TV in standby mode) when media is not being displayed (requires Android v6.0+).

### How to install Video Kiosk as a System App

The HDMI-CEC features require Video Kiosk to be installed as a System App. To install Video Kiosk as a System App, do the following:

1. Use a rooted Android device.
2. Place the Video Kiosk .apk file in the folder /system/priv-app
3. Make sure the Video Kiosk .apk read/write/execute security is set to 644
4. Reboot the device.

For an example of how to do this using the adb tool, refer to [Appendix A – Making Video Kiosk a System App, p. ]

*If you need help installing Video Kiosk as a System App or would like to purchase a TV box with Video Kiosk pre-installed as a System App, please contact us.*

### Turning the TV on

Video Kiosk can be configured to both turn the TV on and to change to the correct HDMI input when an event is scheduled. This feature requires the following:

- The TV supports HDMI-CEC
- The Android device supports HDMI-CEC both in hardware and in software

- The Android device is running Android 5+ or Android 6+
- The Android device is a rooted device
- Video Kiosk has been installed as a System App

To have Video Kiosk turn the TV on and to change to the correct input when a scheduled event begins, follow these steps:

1. Install Video Kiosk as a System App.
2. Configure the Settings  
Go to **Video Kiosk Settings > Schedule > HDMI CEC OneTouchPlay** and check the checkbox.

## Returning the TV to the correct input

In addition to selecting the correct HDMI input when a scheduled event begins, Video Kiosk can be configured to periodically check to ensure the TV is still using the selected HDMI input and to change it back to the correct input throughout scheduled events. (Video Kiosk will only return the TV to the correct input when a scheduled event running.)

This feature requires the following:

- The TV supports HDMI-CEC
- The Android device supports HDMI-CEC both in hardware and in software
- The Android device is running Android 5+ or Android 6+
- The Android device is a rooted device
- Video Kiosk has been installed as a System App

To have Video Kiosk turn the TV on and to select the correct input periodically during a scheduled event, follow these steps:

1. Install Video Kiosk as a System App.
2. Configure the Settings  
Go to **Video Kiosk Settings > Schedule > Send HDMI CEC OneTouchPlay periodically in minutes** and change the frequency with which the OneTouchPlay CEC command will be sent.

## Turning the TV off

Video Kiosk can be configured to turn the TV off when a scheduled event is finished. This feature requires the following:

- The TV supports HDMI-CEC
- The Android device supports HDMI-CEC both in hardware and in software
- The Android device is running Android 6+
- The Android device is a rooted device
- Video Kiosk has been installed as a System App

To have Video Kiosk turn the TV off when a scheduled event is over, follow these steps:

1. Install Video Kiosk as a System App.
2. Configure the Settings  
Go to **Video Kiosk Settings > Schedule > HDMI CEC SendStandby** and check the checkbox.

## Keyboard and Remote Control Features

Video Kiosk supports some keys on keyboards and remote controls. How these buttons are mapped to various remote controls and keyboards is vendor specific.

### Next video

The button used to select the next video corresponds to any of the following:

- KEYCODE\_MEDIA\_NEXT (87)
- KEYCODE\_PAGE\_DOWN (93)

### Previous video

The button used to select the previous video corresponds to any of the following:

- KEYCODE\_MEDIA\_PREVIOUS (88)
- KEYCODE\_PAGE\_UP (92)

## Touch Gesture Control

Video Kiosk supports the following touch gesture controls. To enable this feature Go to **Video Kiosk Settings > Buttons > Enable swiping** and check the checkbox.

### Next media item

The touch gesture used to select the next media item corresponds to any of the following:

- Swipe from right to left

### Previous media item

The touch gesture used to select the previous media item corresponds to any of the following:

- Swipe from left to right

# Gamepad Control Features

Video Kiosk supports some buttons on gamepads. How these buttons are mapped to various gamepads is vendor specific.

## Display button bar

The button used to display the button bar corresponds to any of the following:

- KEYCODE\_DPAD\_DOWN (20)
- KEYCODE\_DPAD\_UP (19)

## Next video

The button used to select the next video corresponds to any of the following:

- KEYCODE\_BUTTON\_R1 (103)

## Previous video

The button used to select the previous video corresponds to any of the following:

- KEYCODE\_BUTTON\_L1 (102)

# Reliability Features

Video Kiosk is designed to provide robust, reliable playback so that operator intervention is not required to handle common playback issues.

For information on Settings, refer to [Configuring the Settings]

## Handling changes to the playback loop

Video Kiosk handles changes in the VideoKiosk folders without requiring a restart. For example, if the media folder contents change, the new selection of media will play the next time the content loop starts. This allows you to remotely update the content without having to stop the device.

## Handling unplayable content

Video Kiosk supports a variety of image and video formats. If there is content in the media folder that is not in a supported format, Video Kiosk will skip it and play the next file. If the file is in a supported format, but unplayable for a different reason (for example an empty or corrupt file), Video Kiosk will skip it and play the next file.

If too many errors occur in a row, Video Kiosk will try to correct the problem by restarting.

- *On a rooted device, and if Video Kiosk is the Home App*, Video Kiosk will reboot Android and Video Kiosk will re-launch and start playing again after reboot
- *Otherwise*, Video Kiosk will exit and re-launch Video Kiosk and start playing again

## Behavior on wake from sleep

Video Kiosk will automatically start playing content upon waking from sleep. When the device wakes, Video Kiosk can play the current file or play from the beginning of the media folder.

- To change this behavior, go to **Video Kiosk Settings > Sleeping > On wake from sleep** and pick an option

## Error Detection Automatic Restart

Video Kiosk keeps track of playback errors. If too many errors occur without an intervening success (30 videos, 300 webpages), the App will automatically restart.

- To turn this feature on |off, go to **Video Kiosk Settings > Launching > Restart on errors**

## Periodic Automatic Restart

Periodically restarting a device is a common way of addressing Android OS and hardware instability issues. Video Kiosk will automatically restart the device or restarts Video Kiosk at the specified intervals. The automatic restart feature is used to restart the device at a configurable interval time. The default behavior is to restart once each night at midnight.

- To turn this feature on |off, go to **Video Kiosk Settings > Launching > Restart Daily**
- To change how often and at what time the restart occurs, go to **Video Kiosk Settings > Launching > Restart Times**
- To test the Restart daily setting now, go to **Video Kiosk Settings > Launching > Test Restart** . On rooted devices, this may trigger a reboot when you exit the settings.

# Access Management

The Access Management features control the access both to the device controls and to the video kiosk software. Restricting the functionality of your device to kiosk functionality is an important part of controlling what content is being shown on your device, especially if you are running an interactive kiosk. Otherwise an inquisitive or malicious user may make changes to your device by pausing the video, decreasing the volume or quitting from your kiosk altogether. To prevent such changes from occurring, Access Management is available.

For information on Settings, refer to [Configuring the Settings]

Video Kiosk provides access management functionality both to limit access to the device controls and to the Video Kiosk settings. Which access management features are supported depends on whether or not the device is rooted, as documented in the following table:

Feature	Non-rooted device	Rooted device
Set Video Kiosk as the Home App	✓	✓
Password protect access to the Video Kiosk User Interface(UI)	✓	✓
Close power button long press menu	✓	✓
Disable volume buttons	✓	✓
Show   Hide media controller	✓	✓
Prevent access to System User Interface (UI)	✓	✓
Using Video Kiosk as the single Launcher	✓	✓

## Access Management recommendations

Because Video Kiosk is used in a variety of different kiosk environments, it is not always necessary to limit access to the device. However, Video Kiosk includes functionality to allow you to limit user access to the device, should that be required. For example, a wall-mounted TV used as a kiosk to play video at the gym would require less access management than an interactive kiosk in the hands of curious teenagers at a museum. Video Kiosk's Access Management features are designed to be flexible so that you can configure access at a level appropriate to the kiosk environment.

Video Kiosk can be run with no Access Management. However, to help ensure your kiosk is used as a kiosk and that it is as reliable as possible, we recommend the Basic Access Management Setup be used in all Video Kiosk installations.

### Basic Access Management Set Up – Setting the Home App

The Home App is the app that an Android device will return to when the Home button is pressed and will launch when Android is restarted. In order to have Video Kiosk automatically restart when

Android is restarted, set it as the Home App. We recommend you always set Video Kiosk as the Home App. To do this, follow these steps:

#### ***On Android 4 and Android 5, if your version has a system setting for the home app:***

1. In the Video Kiosk button bar, press the Video Kiosk Settings button and go to **Video Kiosk Settings > Launching > Open System Settings**
2. Scroll down until you find the Home setting and press Home. A list will display.
3. Touch the radio button to select "Video Kiosk". Touch the "Back" button 3 times to return to the Video Kiosk screen.
4. Verify that Video Kiosk is the Home app. In the Video Kiosk button bar, touch the Home Button. A dialog will display the current Home App. If the current Home App is "Video Kiosk", you have successfully set the Home App. Press "Cancel" to dismiss the dialog. Otherwise, repeat steps 2-4.

#### ***On Android 4:***

1. In the Video Kiosk button bar, touch the Home Button. A dialog will display the current Home App. If the current Home App is "Video Kiosk", press "Cancel".
2. Otherwise, press "Android System". Android displays the "Select a home app" dialog. Android requires you to now select a home app. (If you don't select one, this dialog will pop up every time the device is restarted until you select a home app.)
3. Select "Video Kiosk" as the launcher and press the "Always" button.
4. Verify that Video Kiosk is the Home app. In the Video Kiosk button bar, touch the Home Button. A dialog will display the current Home App. If the current Home App is "Video Kiosk", you have successfully set the Home App. Press "Cancel" to dismiss the dialog. Otherwise, repeat steps 2-4.

#### ***On Android 5:***

1. In the Video Kiosk button bar, touch the Home Button. A dialog will display the current Home App. If the current Home App is "Video Kiosk", press "Cancel".
2. Otherwise, press "Android System". Android displays the "Set Home" dialog. Android requires you to now select a home app. (If you don't select one, this dialog will pop up every time the device is restarted until you select a home app.)
3. Select "Video Kiosk" as the home app. The dialog will dismiss.
4. In the Video Kiosk button bar, touch the Home Button. A dialog will display the current Home App as "Video Kiosk". Press the "Always" button.
5. Verify that Video Kiosk is the Home app. In the Video Kiosk button bar, touch the Home Button. A dialog will display the current Home App. If the current Home App is "Video Kiosk", you have successfully set the Home App. Press "Cancel" to dismiss the dialog. Otherwise, repeat steps 2-5.

#### ***On Android TV 6:***

By default, you are not allowed to change the home App. However, on a rooted device, you can work around this by simply disabling the built-in launcher and Video Kiosk will automatically become the launcher. There are several ways to disable the built-in launcher. The following example shows how to do it using the adb developer tool.

*Example, using the abd developer tool.*

1. First install Video Kiosk.
2. Connect your device to a computer with the developer tools installed
3. From a command line on the computer, enter these commands:

```
adb shell pm hide com.google.android.leanbacklauncher  
adb reboot
```

Video Kiosk will now automatically restart when Android is restarted.

### ***On Android 6 and later if you configure Video Kiosk as the device owner:***

(For an example of how to make Video Kiosk the device owner, refer to Appendix B – Making Video Kiosk the device owner at the end of this manual.)

1. In the Video Kiosk button bar, touch the Home Button.
2. A dialog will display telling you how many launchers are active on the device.
3. Press the button titled **Video Kiosk** to hide all the other launchers on the device.

This is the recommended option on versions of Android that are not rooted.

You can touch the button titled **All** to unhide any other launchers that may be present on the device. After doing this, you may be prompted to choose a launcher when you touch the home button or when the device reboots.

*If you need help disabling the launcher or you wish to purchase a TV Box where Video Kiosk has already been installed as the only launcher, please contact us.*

## **Basic Secure Access Management Set Up**

On a Mac, you use the Finder to manage file and launch applications; on Windows, you use Windows Explorer. The Android OS equivalent is called Launcher and unlike Mac and Windows, Android allows you to have more than one launcher. Video Kiosk is a launcher, allowing you to launch applications from within Video Kiosk itself. And because you can password protect Video Kiosk, you can restrict access to the other applications on the device to only those individuals who know the Video Kiosk password.

To do this, follow these steps:

1. **Set Video Kiosk as the Home App**  
If you haven't already done so, set Video Kiosk as the Home App. (See above.)
2. **Password protect Video Kiosk.** Set the password required to make changes to the kiosk. You can password protect access to Video Kiosk in two ways, by putting a password file in the default location or by setting the password in the App Settings. The password in a password file takes precedence over the password in the Video Kiosk settings. If you delete the password file, the password that was set in the Video Kiosk settings becomes active again.

### *Use the Default Folder Location*

- To use the default location, create a password file with a name that is a SHA512 encoding of someword, **SHA512(someword).password**, and put the password file in the same folder as your media files. Video Kiosk will then use “someword” as the password.  
e.g. A file called  
**36622CA176A6D999D3C2B41D3E002A05B4342C8902DCD98145372C2F0A0B9A0031DC1424FFE825F975A6E390887354E903C460E3E374D95A36CCA2C058B96D7A.password**  
is upload to the media folder. Video Kiosk will then use **dragon** as the password.

As a less secure alternative you can use an MD5 encoding, in which case a file named **8621ffdbc5698829397d97767ac13db3.password** would result in Video Kiosk using **dragon** as the password.

As a totally insecure alternative you can use the word itself, in which case a file named **dragon.password** would result in Video Kiosk using **dragon** as the password.

***If you plan to use this feature, and security is important to you, it is highly recommended that you use a SHA 512 encoding of the password.***

### *Set in the App Settings*

- Go to **Video Kiosk Settings > Security > Set Password** and set the password.

Now, only individuals with the Video Kiosk password can access the other applications on the device.

## **Enhanced Access Management Set Up**

We recommend you implement this Enhanced Access Management Setup for all Video Kiosks where users have physical access to the device. Video Kiosk is designed to limit access to these features to prevent users from controlling the device with these features:

1. **Close power button long press menu.** Configured in the Video Kiosk Settings screen, this feature disables the power button long-press menu.  
Go to **Video Kiosk Settings > Buttons > Close power button long press menu** and check the checkbox.
2. **Disable volume buttons.** Configured in the Video Kiosk Settings screen, this feature disables Volume Dialog, which controls device volume, and additional functionality on some Android devices.  
Go to **Video Kiosk Settings > Buttons > Disable volume buttons** and check the checkbox.
3. **Prevent access to System UI.** Configured in the Video Kiosk Settings screen, this feature kills the system UI on a rooted Android v4 device and uses a pinned App on Android v5 and

later devices.

Go to **Video Kiosk Settings > System UI > Prevent access to System UI** and check the checkbox. When you exit Video Kiosk or choose Video Kiosk Settings, the System UI is restored.

Now, if you have set a password, only users with the Video Kiosk password can access the device controls. For information on Settings, refer to [Configuring the Settings]

## **Advanced Access Management Set Up**

One final step you can take to control access to your device is to remove other launchers from the device. This requires access to the Android Developer Tools or, on a rooted device, special Apps like Titanium Backup. We recommend you get this done by the device manufacturer. For further information, contact us directly.

# Remote Management

Because Video Kiosk was designed to be remotely managed, it supports both sending status reports to a URL and receiving responses so you can integrate Video Kiosk remote devices into a management system. Video Kiosk Android examines the response to the HTTP/HTTPS POST for key value pairs that it will use to update the management interface settings.

A management status POST can be sent:

- based on a time interval
- based on a specific time of day
- any time using the Send Management Report button on the button bar
- Now, if you have set a password, only users with the Video Kiosk password can access the device controls.

For information on Settings, refer to [Configuring the Settings]

## How to enable device status reporting

To set up device status reporting, follow these steps:

1. If you plan to send HTTPS POSTs instead of HTTP POSTs, get an SSL certificate. For more information, see [HTTP vs HTTPS considerations] below
2. In **Video Kiosk Settings >Management** configure the following:
  - Unique Device ID and Zone (optional settings for your convenience)
  - When to report status
  - URL to report status to
  - Content of the Post
3. Video Kiosk devices will now send POSTs to the specified URL.

If status reporting is enabled, a status report will be automatically POSTed 2 minutes after the App launchers.

## HTTP vs HTTPS considerations

To use HTTPS, you must get an SSL certificate from a certificate authority that the Android OS recognizes. Video Kiosk does not support HTTPS POST to sites using self-signed certificates.

If your device does not have a real time clock, it is possible that HTTPS will not work until the time is set from the network. If, for some reason, the network time server is not 100% reliable, you should use HTTP POSTs instead of HTTPS POSTs.

If you plan to send the password to the management server, and security is important to you, you should use a SHA 512 encoding and an HTTPS POST. Passwords set using the Video Kiosk settings are stored as SHA 512 encodings of the password.

## Status Report Post Content: Supported POST Key Value Pairs Sent

The values are encoded using the Android URLEncoder and the UTF-8 character set.

- PHP programmers that use `$_POST` variables should decode these values using the `urldecode()` method.
- PHP programmers that use `$_REQUEST` variables should not decode the values since they have already been decoded.

The following keys will always be included in the POST:

`androidid`

The unique id of the device assigned by the Android OS.

`account5`

If an account has been added to the device, an MD5 encoding of the account.

`cloudid`

The Firebase Cloud Messaging Cloud ID that the management server can use to send push notifications to devices if the management interface is accepting push notifications. If you want to use this feature in a proprietary management interface, contact us for more information.

If the `cloudid` changes, a management POST will automatically be sent so that the management server can update its database with the new `cloudid` for this device.

`uid`

The unique id of the device as assigned by you. By default this is the unique id assigned by the Android OS but you can change it to be any string that you like.

`version`

The Video Kiosk version name like *4.1.0.150529*.

`osversion`

The version of Android OS on the device.

`date`

The UTC date on the device at the time the POST was sent in the form `YYYY-MM-DD HH:MM:SS +HH:MM`. An example is `2016-08-07 16:59:49 +00:00`.

The following keys can be included in the POST by selecting them in the Management Settings (**Video Kiosk Settings > Management > Management Post Content**)

isplaying

A flag that will be true if a video is playing or false otherwise. Be careful interpreting the value since it can be false if an image file is being displayed or if one video has ended and its last frame is being displayed while the next video is preparing to play but has not yet rendered its first frame.

currentmedia

The name of the currently playing video or displayed image file. By default this key value pair is enabled.

folderpath

The path to the currently selected media folder. Please note that if you enable this key, then the names of the media items in the foldercontent array will be relative to this path and if you do not specify this key then the names will be the full path.

foldercontent

The names of all the media items in the currently selected media folder.

password

The password assigned to the Video Kiosk using the someword.password file. You can use this to authenticate the Video Kiosk POST. If you are sending this data, you should use HTTPS instead of HTTP, since it's not a good idea to send the password in the clear.

zone

The zone of the device as assigned by you to create any grouping of devices that you would like.

isroot

A flag that will be true if the device has been granted root access or false otherwise.

uptime

The time, in seconds, since the last device reboot

filesize

An array of file sizes that matches the array of file names. [comma-separated list of values]

emptyfiles

An array that lists any files that are empty (probably due to a network error during file synchronization with your cloud service). [comma-separated list of values]

widgetsenabled

A flag that will be true if widget areas are enabled or false otherwise.

areabwidgets

An array that lists the widgets configured in widget area B [comma-separated list of values]

areacwidgets

An array that lists the widgets configured in widget area C [comma-separated list of values]

areadwidgets

An array that lists the widgets configured in widget area D [comma-separated list of values]

onetouchplay

An array that lists first a flag that is set to true if the onetouchplay feature is enabled and then the Onetouchplay interval from the settings [comma-separated list of values]

sendstandby

A flag that will be true if the sendstandby feature is enabled

interactions

An array of key value pairs that lists the identifying string and interaction count for each hotspot for which the interaction count has been set in the Hotspot Overlay .XML. For hotspots without an identifying string, interactions will not be counted or reported. {comma-separated list of key value pairs where key=value}

location

The location of the device as provided by a GPS latitude and longitude pair OR the postal address of the device entered as a human readable string

batterypercent

A floating point representation, for example 47.4, of the current battery charge as a percent.

batteryplugged

A string that indicates both if and how the battery is being charged. The value will be one of the following: AC | USB | WIRELESS | NONE

Whenever you change the settings, Video Kiosk will restart the interval time. If you have enabled the time of day status POST, Video Kiosk will synchronize the interval time with the time of day when that POST is sent.

## Responses: Supported POST Key Value Pairs in Response

The management interface supports responses by the management system to management status POST, such that the management settings on the device can be changed by the management system.

There are two management protocols (protocol 0 and protocol 1) and they use different responses. Protocol 0 is a legacy protocol and should not be used for new management application development. If you are developing a new management application, use protocol 1.

### Protocol 0 Response (deprecated)

Response is a newline-terminated, optionally url-encoded, list of key value pairs.

The management system may respond with a list of key value pairs as follows:

```
Key1=value1\n
key2=value2\n
...
key3=value3\n
```

\n means newline as defined in the PHP scripting language.

The supported key value pairs are:

```
uid=some id
zone=some zone
reporttime=HH:MM (where 0 >= HH <= 23 and 0 >= MM <= 59)
reporttimeenabled=true|false
intervaltime=NNNN (where NNNN > 0)
intervaltimeenabled=true|false
postkey_isplaying=true|false
postkey_currentmedia=true|false
postkey_foldercontent=true|false
postkey_password=true|false
postkey_zone=true|false
postkey_isrooted=true|false
postkey_folderpath=true|false
postkey_widgets=true|false
postkey_hdmicec=true|false
postkey_interactions=true|false
postkey_location=true|false
postkey_battery=true|false
setdate=YYYY:MM:DD:HH:MM:SS
```

If the \n characters are omitted, Video Kiosk will ignore the key value pairs. You MUST include the newline character for every key value pair.

## Protocol 1 Response

Response is a JSON Map where the key is used to access the value.

The management system may respond with a list of key value pairs as follows:

```
{"key1":"value1","key2":value2 ...,"keyN":"valueN"}
```

The supported key value pairs are:

```
uid:"some id"
zone:"some zone"
reporttime""HH:MM" (where 0 >= HH <= 23 and 0 >= MM <= 59)
reporttimeenabled:"true|false"
intervaltime:NNNN (where NNNN > 0) – NOTE: no quotes, its not a string
intervaltimeenabled:"true|false"
postkey_isplaying:"true|false"
postkey_currentmedia:"true|false"
postkey_foldercontent:"true|false"
postkey_password:"true|false"
postkey_zone:"true|false"
postkey_isrooted:"true|false"
```

```
postkey_folderpath:"true|false"  
postkey_widgets:"true|false"  
postkey_hdmicec:"true|false"  
postkey_interactions:"true|false"  
postkey_location:"true|false"  
postkey_battery=true|false  
setdate:"YYYY:MM:DD:HH:MM:SS"
```

Here is an example of a JSON Map response:

```
{"intervaltimeenabled":"true","intervaltime":60,"postkey_isplaying":"false","postkey_currentmedia":"true","postkey_foldercontent":"true","postkey_password":"true","postkey_zone":"true","postkey_isrooted":"true","postkey_uptime":"true","postkey_filesize":"true","postkey_emptyfiles":"false","postkey_folderpath":"true","postkey_widgets":"true","postkey_interactions":"true","postkey_location":"true","uid":"Metrotown"}
```

## Protocol 0 and 1 Response Notes

If both **intervaltimeenabled** and **reporttimeenabled** are set to false, you will stop that device from reporting status. If you do that via a management POST response then management posts can only be restarted manually from the device.

Setting the date and time on a device requires either that Video Kiosk is in /system/priv-app or that the device is rooted. HTTPS requires the date and time to be set correctly so if you want to set the date and time on a device where the time has reverted to the base time on the device you MUST use HTTP for management posts.

If a management POST response changes the current settings on the device, it will immediately send another POST to the management URL

## Example of Use

### Sending an email containing POST Data and set device time - PHP Script

The following PHP script can be used to send an email containing all the POST data to whatever email account you would like. You just need to change the values of \$to\_email, and \$from\_email and then put the script on your web site in a file named something.php. In addition, if the UTC date and time on the device is more than 30 seconds different than the UTC date and time on the server, the time on the device will be set to match the time on the server (this requires that Video Kiosk is in /system/priv-app or that the device is rooted)

```
<?php  
date_default_timezone_set("UTC");  
$to_email = 'you@yourdomain.com';
```

```

$from_email = 'do-not-reply@yourdomain.com';
$subject = 'VidoeKiosk Status';

$headers = 'From: ' . $from_email . "\r\n" .
           'X-Mailer: PHP/' . phpversion();

foreach ($_POST as $key => $value)
{
    $message .= htmlspecialchars($key) . "/" . htmlspecialchars(urldecode($value)) . "\n";
}

if (isset($_POST['date']))
{
    $date = htmlspecialchars(urldecode($_POST['date']));

    $_format = 'Y-m-d H:i:s P';
    $_date = date_create_from_format($_format, $date);

    $_serverDate = date_create();

    $message .= "\nServer UTC Date = /" . date_format($_serverDate, 'Y-m-d H:i:s P') . "/";

    $_clientTS = date_timestamp_get($_date);
    $_serverTS = date_timestamp_get($_serverDate);
    $_interval = abs($_clientTS - $_serverTS);

    $message .= "\n\nDifference in seconds = /" . $_interval . "/";

    if ($_interval > 30)
    {
        echo "setdate=" . date_format($_serverDate, 'Y:m:d:H:i:s') . "\n";
    }

    mail($_to_email, $subject, $message, $headers);
}

?>

```

# Configuring the Settings

If a setting is not supported on a particular device, the setting will not be displayed. For example, if a setting enables a feature that requires Android v6 or later, the setting will not be displayed on a device running Android v5.

For more information on how to find the settings, refer to the [Installing and Configuring] section of this document.

## Settings – Cloud Download

### *Cloud drive info*

Display a summary of Cloud Drive download settings.

### *Enable downloading*

Enable or disable downloading from Cloud drive. Default is off.

### *Download interval*

Set the frequency, in hours, with which Video Kiosk will check for and download changes from the selected Cloud Source folder. Default is 1 hour.

### *Path to local folder*

Set the local path, on the device, to which the selected Cloud Source folder will be downloaded. Default <path-to-external-storage>/videokiosk

### *WebDAV folder*

If you're using a service that supports WebDAV, choose the location of the Cloud Source folder and provide your credentials to access the folder

### *Nextcloud folder*

If you're using Nextcloud, choose the location of the Cloud Source folder on Nextcloud

### *Personal Cloud folder*

If you're using your own web site and have installed our open source Personal Cloud PHP script, choose the location of the Cloud Source folder and provide your credentials to access the folder

### *Google Drive folder*

The integrated google drive download has been deprecated. If you want to use google drive we recommend Autosync Google Drive configured in Mirror (download only mode). You can get more information on this third party App here:

<https://play.google.com/store/apps/details?id=com.ttxapps.drivesync>

There are, of course, many Apps dedicated to downloading a folder from a cloud service and you can use any of those Apps, as you prefer, with Video Kiosk. The only requirement is that the folder exists on the local device.

**Note: You cannot choose only one of the available WebDAV, Nextcloud, or Personal Cloud folders.**

## Settings - File and Folder

### *Enable cache*

Video Kiosk can cache the list of file names from the selected Media folder in memory. Video Kiosk reloads the memory cache from the SD card only if the content in the Media folder changes. You can, if you prefer, turn this memory cache off and Video Kiosk will read the file names from the SD card each time it starts the loop. Default is on.

### *Sort by*

Pick the Sort Order of the Media Folder. The default is "Path of first playlist found"

#### **Options:**

*Path or first playlist found* – Sorts by path or by the first playlist found (if there is one)

*Path* – Sorts by full path name

*File name excluding path* – Sorts by Filename, excluding path

*Random shuffle* – Sorts in a random order

*Playlist* – Sorts by the playlist you pick. All playlists found in the Playlist folder will be displayed.

Pick one.

### *Sub-folder depth*

Pick from the menu to limit the depth of nested sub-folders that will be played. Default is 5.

### *Limit playback to documented formats*

Limit the media files that will be played to 3gp, mp4, ts, webm, mkv, jpg, png, gif, bmp, webp.

When this is set, media files of other types will be skipped. The default is off.

### *Show playback errors*

When this is set, a brief error message will be displayed if a video cannot be played and then the video will be skipped. When this is not set, a video with playback errors will just be skipped. The default is off.

### *Path to playlist folder*

Choose the folder that contains your playlist m3u8 files. Choose the folder that contains your schedule file. (Not required if using Standard Folder Layout or One Folder Layout. Required if using Custom Folder Layout.)

This is only required when using playlists and using the Custom Folder Layout.

## Settings - Timers

### *Image display time*

How long (in seconds) images will be displayed if the file name does not include a timeout value.

Default is 30 seconds. When this is set to the value "0", Video Kiosk will pause on the image until the display is touched.

*Image fade time (Android v4.2 and later)*

How long (milliseconds) image will take to fade in and out. Default is 0 seconds.

*Web fade time (Android v4.2 and later)*

How long (milliseconds) web pages will take to fade in and out. Default is 0 seconds.

*Video fade time (Android v4.2 and later)*

How long (milliseconds) video will take to fade in and out. Default is 0 seconds. NOTE: Setting this time greater than 0 can result in choppy video playback on low-end devices. If you experience choppy video playback, set this to 0.

*Finish playback first*

By default, Video Kiosk will wait for the current media item to finish playing and then restart. If you prefer, you can disable this option and Video Kiosk will restart exactly on the hour without waiting for the current media item to finish playing.

*Schedule check seconds*

How often (in seconds) Video Kiosk will check for changes to the schedule. The default is 30 seconds.

## Settings - Launching

*Launch on startup*

Automatically launch Video Kiosk when the device is restarted. If you enable this setting, the standard launcher will open and then Video Kiosk will open. Alternatively, you may use the Video Kiosk button bar to set Video Kiosk as the home / launcher App.

*Restart on errors*

Video Kiosk keeps track of playback errors. If too many errors occur without an intervening success (30 videos, 300 webpages), the App will automatically restart when this setting is on. The default is “on”.

*Exit and relaunch*

Automatically exit Video Kiosk when the media loop finishes playing and relaunch Video Kiosk again after a configurable number of minutes. This is a specialized feature used with devices that are distributed as Android TVs that display a periodic interstitial advertisement.

*Restart Daily*

Automatically restart the device, at the times specified in the “Restart Times” setting. On a rooted device, this may trigger a reboot. Behaviour depends on whether the device is rooted or not and whether or not Video Kiosk as set as the home app, as documented in this table:

Option:	Rooted Device	Non-rooted Device
<i>Do not restart</i>	Does not restart daily	Does not restart daily

Option:	Rooted Device	Non-rooted Device
<i>Exit and relaunch the App</i>	Exits and relaunches Video Kiosk	Exits and relaunches Video Kiosk
<i>Reboot the device (requires root)</i>	<p><i>Video Kiosk is the Home App:</i> Reboot Android and relaunches Video Kiosk</p> <p><i>Video Kiosk is not the Home App:</i> Exits and relaunches Video Kiosk</p>	Exits and relaunches Video Kiosk

### *Restart Times*

By default, if you enable Restart Daily, Video Kiosk will restart once per day at midnight. You can change the restart time to a different hour of the day or you can choose to restart multiple times daily (on the hour) by selecting multiple restart times.

### *Test restart daily*

Test the Restart daily setting now. On rooted devices, this may trigger an immediate reboot.

### *Launch another App*

Launch a different installed App. Use this to launch apps if you have disabled the Launcher. This is a convenience on a device that has multiple launchers, allowing you to launch apps without having to reconfigure the device with Launcher 3 as the Home App. On a device with only one launcher – Video Kiosk – this allows you to launch apps, provided you have the password to access the Video Kiosk settings, providing you with the functionality you required while effectively prevents malicious users from starting another app on your kiosk.

### *Open System Settings*

Open the system settings now. Use this to access the system settings if you have disabled the Launcher. This is a convenience on a device that has multiple launchers, allowing you to launch apps without having to reconfigure the device with Launcher 3 as the Home App. On a device with only one launcher – Video Kiosk – this allows you to access the system settings, provided you have the password to access the Video Kiosk settings, providing you with the functionality you required while effectively prevents malicious users from accessing the system settings on your kiosk.

## Settings - Buttons

### *Show button bar when App launches*

By default, as a hint that it exists, the button bar is displayed for 30 seconds after the Video Kiosk App launches. In some cases, you may not want the button bar to display. To prevent the button bar from being displayed for 30 seconds after the Video Kiosk App launches, disable this setting.

#### *Show button bar pattern*

By default, whenever the display is touched, the button bar will appear. In some cases, you may want to use a specific pattern of touches to cause the button bar to appear. You can set that pattern by using this setting.

1. Enter a series of digits where
  - 0 means a touch anywhere on the screen,
  - 1 means a touch in the top left,
  - 2 means a touch in the top right,
  - 3 means a touch in the bottom left
  - 4 means a touch in the bottom right

*Example:* if you want the button bar to display when the display is touched in this order: top left, top right, bottom left, bottom right, you would enter 1234 as the pattern.

#### *Play touch tones*

If you have configured a **Show button bar pattern**, then turn on this setting to play touch tones when you tap the four screen quadrants so that there is audible feedback when a touch is detected. The default is “off”

#### *Close power button long press menu*

After 30 seconds or Video Kiosk restart, close power button long-press menu. This prevents unwanted users from accessing the Power button’s long press menu.

#### *Disable volume buttons*

Disable the buttons on the Volume Dialog. This includes disabling the Volume up and Volume down buttons, to prevent unwanted users from changing the volume setting on your kiosk. On some Android devices, there are additional settings available on this dialog. These are also disabled when this is set.

## Settings – Interactive Playback

#### *Mode*

Set the playback mode. The default is “Play next”.

#### **Options:**

*Play Next* - plays the next media item when the current media item finishes

*Play previous* – plays the previous media item when the current media item finishes

*Pause* – pauses playback when the current media item finishes. Use swipe or keyboard event to play the next or previous media item.

*Play again* – play the same media item again in a loop. . Use swipe or keyboard event to play the next or previous media item.

NOTE: For backward compatibility, if **Video Kiosk > Settings > Timers > Image display time** is set to 0, then the playback mode **Play next** will be used regardless of this setting. The image will display until you tap the screen and then the next media item will play.

*Show media controller (Android v4.2 and later)*

Show or hide the media controller for videos. By default, when video is playing there is no media controller and so the user cannot interact with the video. If you enable this option a media controller will overlay the video and the user will be able to scrub, fast forward, and even skip videos.

*Enable swiping*

Enable using swipe gesture to skip media items in playback. The default is “off”.

## Settings - System UI

*Prevent access to System UI*

On a rooted Android v4 device, this kills the system UI to prevent the system and navigation bars from being accessed by users.

On an Android v5 and later device, this uses App pinning to restrict users from accessing anything other than the Video Kiosk App. When you exit Video Kiosk, or enter the Video Kiosk settings, the System UI access is restored.

To use App pinning,

1. Set Video Kiosk as the device owner. This must be done prior to adding any user accounts to the device. Setting an App as a device owner can be done in several ways. (For an example of how to make Video Kiosk the device owner, refer to Appendix B – Making Video Kiosk the device owner at the end of this manual.
2. Video Kiosk > Settings > System UI > Prevent access to System UI

## Settings - Sleeping

*On wake from sleep*

Resumes video playback when the device wakes from sleep. The default is to start playing from the beginning of the Media folder.

**Options:**

*Play Current from Pause* - Play starting from where it was playing when the device went to sleep

*Play current from Start* - Play starting from the beginning of what it was playing when the device went to sleep

*Play First from Start* - Play starting from the beginning of the media folder

## Settings - Schedule

*Path to schedule folder*

Choose the folder that contains your schedule file. (Not required if using Standard Folder Layout or One Folder Layout. Required if using Custom Folder Layout.)

*Use a calendar or schedule*

Choose the schedule that will control the device. The schedule list displays all the Local XML schedule files, Android Calendars and Google Calendars it knows about.

#### *Monitor battery charge*

Monitor the battery charge state and change playback when battery is low.

##### **Options:**

*Stop playing media if the device is not charging* – The device will monitor the battery charge state and stop playing media if the device is not charging.

*Monitor the battery level* – The device will monitor the battery charge state and stop playing media if the device charge is below the “Low battery percent” threshold you specify and return the display to normal when device charge is above “High battery percent” threshold.

#### *Detect motion*

Use the camera to detect motion.

##### **Options:**

*Use the camera to detect motion* – The device will use the camera to detect motion and start playing. It will stop playing media if no motion is detected for the specified timeout.

*Motion detection timeout minutes* – The device will stop playing media if no motion has been detected after the specified number of minutes. Default is 5 minutes.

#### *Display state*

Determines what happens when there is a schedule event. Choose the state of the display based on events.

##### **Options:**

*Always on* - Events will not change the display state

*On for events, otherwise off* – Display will display the screen for schedule events and will dim the screen otherwise. (Example: At the gym, use the schedule to turn the display on during classes but otherwise turn it off.)

*Off for events, otherwise on* – Display will turn off for schedule events and will turn on otherwise. (Example: At the Nursing Home, use the schedule to turn the display off during the event “night time” but otherwise leave it on.)

#### *Display clock*

Choose the device behaviour when no schedule event occurs. The default is to display a blank screen. If you set this to “on”, the device will display the time of day and the calendar date.

NOTE: To switch between a 12 hour AM/PM clock and a 24 hour clock, use the System Settings for the device.

#### *Clock color*

Choose the text color for the clock.

#### *HDMI CEC OneTouchPlay*

On Android 5.1+, where Video Kiosk has been installed as a System App in /system/priv-app, send an HDMI CEC OneTouchPlay message to keep the TV turned on the device input selected when the schedule indicates media should be playing.

#### *OneTouchPlay Interval*

How often, in minutes, to send OneTouchPlay messages. Any non-zero value means send the OneTouchPlay message every “interval” minutes. A zero value means do not send OneTouchPlay based on an interval.

#### *HDMI CEC SendStandby*

On Android 6.0+, where Video Kiosk has been installed as a System App in /system/priv-app, send SendStandby to turn the TV off (put it in standby mode) when the schedule indicates media should stop playing.

#### *Enable/Disable HDMI*

This is an experimental feature and is not meant to be used.

#### *Toggle HDMI*

This is an experimental feature and is not meant to be used.

#### *Path to SysFS HDMI folder*

This is an experimental feature and is not meant to be used.

## Settings - Location

#### *Device location*

Use the location provided by the device GPS (Current Location) OR use the postal address of the device entered as a string. This is the location that will be reported on the Management Interface

#### *Path to location folder*

Choose the folder that contains your location files (XML, JSON). (Not required if using Standard Folder Layout or One Folder Layout. Required if using Custom Folder Layout.)

#### *Use Geofence locations*

Turn Geofences on or choose a location / geojson file to use

#### *When to check Geofences*

When to check the current location against the Geofences

#### **Options:**

*Prior to each loop* – Check current location before playing the media loop

*Prior to each media item* – Check current location before playing each item in the media loop

## Settings - Overlays

#### *Path to overlay folder*

Choose the folder that contains your overlay files (png). (Not required if using Standard Folder Layout or One Folder Layout. Required if using Custom Folder Layout.)

#### *Overlay file*

Choose a png overlay file in that folder with a file name ending in .overlay.png

#### *Overlay image scale type*

How Android handles overlay images that are a different size than the screen is determined by the scale type setting. Typically the default, Fit Center, is sufficient but you may have a specific need to change it. For more information on ImageView scale types refer to the Android documentation.

#### **Options:**

*Matrix*

*Fit XY*

*Fit Start*

*Fit Center*

*Fit End*

*Center*

*Center Crop*

*Center Inside*

#### *Hide overlay*

Used to hide the overlay if the screen is not touched for more than 15 seconds. When the screen is touched, if the overlay is visible it will hide and if the overlay is not visible it will be displayed again. If this setting is not enabled, the overlay will never be hidden. The default is “off”

#### *Highlight hotspots*

Used to help test and debug your .xml hotspot file. Turn this on to highlight your hotspots on your screen so that you can ensure they are in the correct place and functioning as expected. Remember to turn this feature off before releasing the device to users.

#### *Web inactivity timeout*

The timeout for an inactive device in WebView, after which Video Kiosk returns to playing the media loop. Default is 30 seconds.

If you set the value to 0, the WebView will not timeout and the only way to get back to the media playback loop is for the user to use the Back navigation or for the Javascript method `window.VideoKiosk.closeWebView()`; to be invoked.

If you turn WebView navigation off, and set the Inactivity time to 0 then you must use Javascript and invoke `window.VideoKiosk.closeWebView()`; to return to the media playback loop.

#### *Show WebView navigation*

By default, when a WebView opens a back button is present and as navigation occurs a backwards and forwards page stack is maintained. This setting allows you to turn off navigation. Default is on.

If you turn navigation off, the user will return to the media loop playback when the Webview inactivity timer expires.

If you turn navigation off and set the Web inactivity timer value to 0, the user will not be able to navigate back from the WebView and the only way to get back to the media playback loop is for

the inactivity timer to expire or for the Javascript method `window.VideoKiosk.closeWebView()` to be invoked.

If you turn WebView navigation off, and set the WebView inactivity time to 0 then you must use Javascript and invoke `window.VideoKiosk.closeWebView()` to return to the media playback loop.

## Settings – Background

### *Path to background folder*

Choose the folder that contains your background file(s) (png, jpg, mp3). (Not required if using Standard Folder Layout or One Folder Layout. Required if using Custom Folder Layout.)

### *Background file*

Choose a png or jpg background file in that folder

### *Next Background Time*

Choose how long, in minutes, until the next background file loads. Only set this if you want the background image to change.

### *Audio URL*

Enable or disable background audio playback and choose the URL to the background audio stream or local file.

## Settings - Management

### *Protocol version*

The management protocol version supported by Video Kiosk. The default value is 0.

Possible values are:

0 – legacy (INI key=value with newline delimiter) responses

1 – JSON encoded responses

NOTE: The default value is 0 for backwards compatibility but you should use protocol version 1 and develop your management server using JSON. Protocol version 0 is deprecated and no new features will be added to protocol version 0.

NOTE: Older versions of Video Kiosk will not send the protocol version. If you are developing your own management system, ensure that your management server assumes a protocol of 0 and sends legacy responses in the case of a Protocol version null value.

### *Device ID*

A unique ID for this device (user specified)

### *Device Zone*

A zone for this device (user specified)

#### Report Interval

How often (minutes) to report status, enable or disable

#### Report Time

Specific time of day to report status, enable and disable

#### URL

Where to POST device status

#### Management POST Content

The keys and values that will be in the POST. To send the key value pair, check its checkbox here.

- Is video playing currently? (isplaying= true | false)
- What media is playing? (currentmedia= filename)
- What media folder was selected (folderpath=path)
- What is in the folder? (foldercontent= [an array of file names])
- Device Password (password=device's password)
- Device zone (zone=user configured zone)
- Root access granted (isroot= true | false)
- Time since last reboot (uptime = the length of time the system has been up)
- File size (filesize=[an array of file sizes in bytes])
- Empty files (emptyfiles=[an array of file names for all files with a size of 0])
- Widget information (widgetsenabled=true|false, areabwidgets=[an array of names], areacwidgets=[an array of names], areadwidgets=[an array of names])
- Which HDMI-CEC messages are enabled (sendstandby=true|false, onetouchplay=true|false)
- Count for each hotspot interaction, if counting configured (interactions)
- Device Location (location)

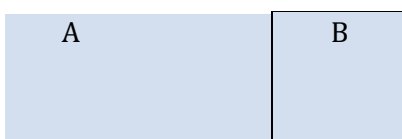
#### Accept push notifications

Enable this setting if the Video Kiosk client should accept push notifications from Google's Firebase Cloud Messaging. Push notifications can be sent to Video Kiosk from the Management server to request the device immediately POST its status. If you are developing a proprietary management server and would like to send push notifications to Video Kiosk please contact us.

## Settings - Widgets

#### Enable Widgets

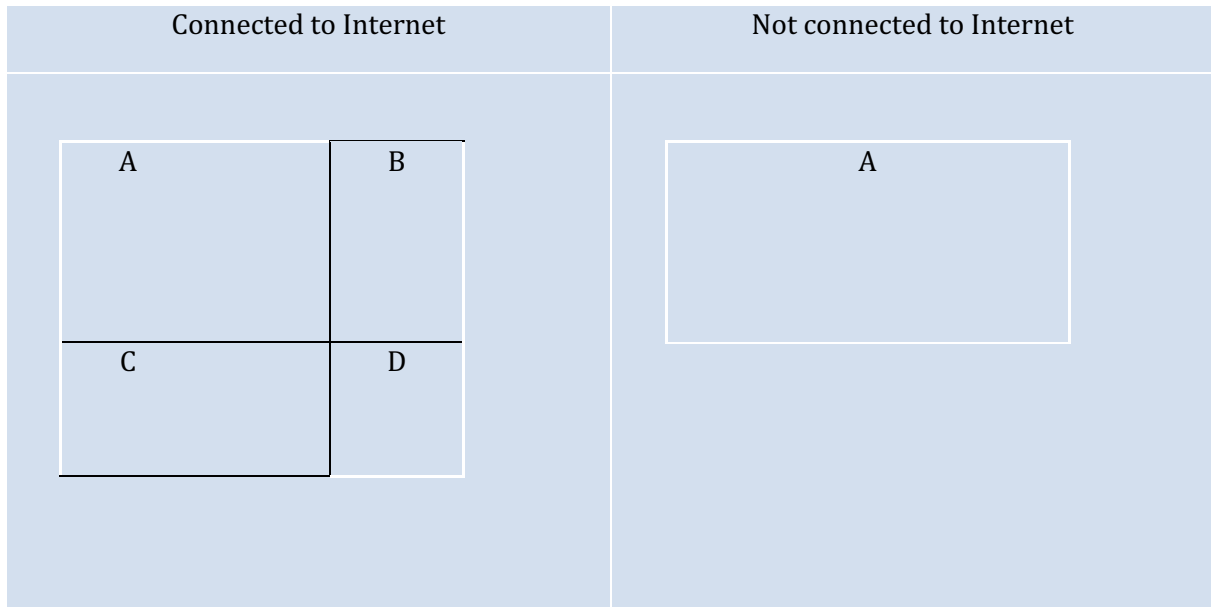
Enable widgets for this device. If you change this setting, Video Kiosk will display a 4 panel layout, including media content in Area A and widgets in areas B, C and D. If you don't configure widgets for areas B, C, D, background will be displayed. If you haven't configured a background, a black screen will display in those areas.





*Monitor Internet*

Prior to playing the next media item, check the state of the connection to the Internet. If the device is not connected to the Internet then hide widget areas B, C, and D. If the device is connected to the Internet show all the widget areas.



*Draw frame on Area A*

Draw a frame around Area A

*Choose Area B Widget*

*Choose Area C Widget*

*Choose Area D Widget*

Pick and configure widgets | URLs for Area B, C and D and choose display time (in minutes). Default is 60 minutes.

*Area A Padding*

*Area B Padding*

*Area C Padding*

*Area D Padding*

Padding allows you to line widgets up horizontally and vertically. Use this setting to set padding, in dp, around widget area – Left, Right, Top and Bottom. Default is 13dp for Area A and 0 dp for Areas B, C and D, which displays with the padding determined by the widget. To decrease the

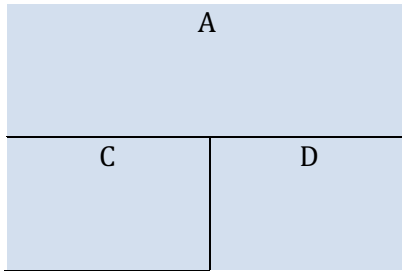
space between widgets, set the padding to a negative (-) number. To increase the padding, increase the padding positive number values.

*Arrange in rows*

Arrange widgets in two rows instead of two columns. The default is to arrange widgets in two columns.

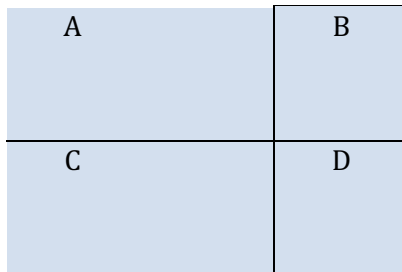
*Horizontal layout*

Layout weights allow you to change the size of the areas on the screen. You can use the horizontal layout weights to alter the horizontal width of each area. For example on a 1080p display in landscape orientation with a resolution of 1920x1080 you could create a layout with Area A taking the full width of the screen, Area B hidden, and Areas C and D taking 50% of the screen using the horizontal width settings of; Area A: 1920, Area B: 0, Area C: 960, and area D: 960. The resulting layout would look like this:



*Vertical layout*

Layout weights allow you to change the size of the areas on the screen. You can use the vertical layout weights to alter the vertical height of each area. For example on a 1080p display in landscape orientation with a resolution of 1920x1080 you could create a layout with Area A and B taking the top half of the screen, and area C and D taking the bottom half of the screen using the vertical height settings of; Area A / B: 540, Area C / D: 540. The resulting layout would look like this:



## Settings - Security

*Set Password*

Set the password required to make changes to the kiosk.

**Valid Values:** Any valid Android password

## Settings - License

### *Deactivate*

Deactivate the license on this device. This enables volume license holders to migrate licenses between devices.

## Settings - View Options

### *Request best resolution*

On Android v6 and later Apps can request that the display resolution be upgraded to 4K rendering on compatible hardware. If you enable this setting, Video Kiosk will look for the best resolution available on the device and request that resolution in an effort to be upgraded to 4K resolution. By default, this setting is off and Video Kiosk will not request that the display be upgraded. (If your device is not running Android v6 or later, this setting will not be displayed.)

### *Screen orientation*

Set the screen orientation of the media playback screen to either the default for the system or a specific orientation based on how you plan to use the display.

#### **Options::**

Default – The default screen orientation set by the manufacturer of your device

Landscape – screen width is wider than is screen height

Portrait – screen height is taller than screen width

Reverse Landscape – Landscape orientation with device turned 180 degrees from landscape

Reverse Portrait– Portrait orientation with device turned 180 degrees from portrait

### *Use textured video views*

Turn on or off textured video view. If textured video views are on the 2D rendering pipeline is used and if textured views are off the 3D rendering pipeline is used. Typically this setting is used to eliminate the black flash problem on amlogic-based devices.

### *Pause playback*

Configure Video Kiosk to invoke `Pause()` rather than `StopPlayback()` when the end of a video is reached. This setting can result in playback errors on some devices and so should only be enabled if Use textured video views does not completely eliminate black flashes between videos. Typically this setting is used to eliminate the black flash problem on Amlogic-based devices. Very few devices require this setting to be enabled.

### *Preload URLs*

Turn on or off URL preloading. If Preload URLs is on Video Kiosk will load URL files in the background when the previous item is being displayed. Typically this setting is used to eliminate visible loading of web content. However, if your web content contains audio, and this setting is enabled, the web page audio may start playing while the previous item is still being displayed. To avoid this audio issue, disable Preload URLs to avoid the audio issue.

#### *ImageView scale type*

How Android handles images that are a different size than the ImageView is determined by the ImageView scale type setting. Typically the default, Fit Center, is sufficient but you may have a specific need to change it. For more information on ImageView scale types refer to the Android documentation.

#### **Options:**

- Pan and Zoom (Ken Burns) effect
- Fit XY
- Fit Start
- Fit Center
- Fit End
- Center
- Center Crop
- Center Inside

#### *Resize images to fit*

Turn on or off automatic image resizing. If Resize images to fit is on, then when a display is rotated, Video Kiosk will automatically resize the image to the largest possible size while maintaining the aspect ratio. You may have a specific need (for example you are using a landscape display in portrait orientation) to turn off automatic image resizing.

#### *Dither JPG images*

Turn on or off JPG dithering. This setting can affect the quality of your JPG images on older versions of Android

#### *Use EXIF Orientation*

Turn on or off the use of the EXIF Orientation META data for image rotation. When this setting is enabled, Video Kiosk will automatically rotate images based on the EXIF data embedded in the image. The default is "on".

#### *Max image megapixels*

Set the maximum number of megapixels. If the image is larger than this, it will be reduced in size by 50%. Historically Android has had problems displaying large images and can crash with "out of memory" errors. If you are getting "out of memory" errors, you can reduce the size of this setting to make them go away. If you are not having "out of memory" errors, you can increase the size of this setting to display higher resolution images.

#### *Show media name*

Turn on or off the display of the media item name. When this setting is enabled, Video Kiosk will display either the title provided using filename meta data in the filename or the file name if a title is not provided. The default is "off".

## Settings - Legacy

### *First frame render time*

How long (1/100<sup>th</sup> seconds) the first video frame takes to render  
Older Android devices can flicker as they attempt to render the first frame of video. This flickering can be eliminated by increasing the first frame render time.

### *Stretch video*

Stretch video horizontally. Typically used to stretch legacy, 960x720 video to new standard 1280x720, in order to eliminate black bands.

# Troubleshooting

Here are some troubleshooting tips.

## The Android Device

### Using uncertified devices

In 2019, Google began blocking uncertified devices (like developer boards) from access play services unless the device Google Services Android ID (GSAID) is registered with Google. To facilitate the registration of uncertified boards, Video Kiosk displays the GSAID on a button that will take you directly to the registration page. For more information, refer to Appendix E – Registering uncertified devices.

### Screen Burn-in

Screen burn-in is a discoloration of areas of the display that is caused by non-uniform use of the pixels. This commonly happens with displays that have the same non-moving image displaying for a prolonged period of time. [[https://en.wikipedia.org/wiki/Screen\\_burn-in](https://en.wikipedia.org/wiki/Screen_burn-in)]

If you display a background image behind your content loop, the background image has the potential to create burn-in. To reduce the risk, follow these suggestions:

- Use a uniform-colored background image.
- If you are using a non uniform colored background image, rotate through more than one. (For instructions on how to do this, refer to “Controlling the Display – Background Images”.)

# Appendix A – Making Video Kiosk a System App

If you have access to the Android developer tools, and your device shell can become root, then you can use this method from the command prompt to install Video Kiosk as a System App.

## 1. Remount /system as read-write

```
adb root
adb start-server
adb shell
  su
  mount -o rw,remount /system
  exit
```

## 2. Copy the apk to /system/priv-app

```
adb push VideoKiosk-release.apk /system/priv-app/VideoKiosk-release.apk
1
```

## 3. Set the App permission

```
adb shell
  su
  chmod 644 /system/priv-app/VideoKiosk-release.apk
  mount -o ro,remount /system
  exit
```

## 4. Reboot the device

```
adb reboot
```

# Appendix B – Making Video Kiosk the device owner

You can make Video Kiosk the device owner from the command prompt on your computer

- If you have access to the Android developer tools (installed on your computer) and
- If your device is running Android v5 or later

*Note: On some Android v6 devices Video Kiosk will pin itself and then Android will immediately unpin Video Kiosk. This typically means there is a setting (Device Settings > Security >Screen pinning) that you need to turn on. Once that setting is enabled, everything will work as described.*

Verify that your device allows the device admin feature by doing the following:

1. From a command line on the computer, enter this command:

```
adb shell pm list features | grep device_admin
```

You should get a response back that looks something like this:

```
feature:android.software.device_admin
```

If your device does not support setting the device owner you will get a blank line back

In the case where your device does not support the software.device\_admin feature you may be able to enable it by doing the following:

1. Create a text file named **android.software.device\_admin.xml**
2. Edit the file and put the following content in it:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (C) 2014 The Android Open Source Project
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
    http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<permissions>
  <feature name="android.software.device_admin" />
</permissions>
```

3. From a command line on the computer, enter these commands:

```
adb root
adb remount
adb push android.software.device_admin.xml /system/etc/permissions/
adb reboot
```

4. When the device has finished rebooting, verify that your device allows the device admin feature by following the steps outlined above

If, and only if, you are able to verify that your device allows the device admin feature, perform the following steps to make Video Kiosk the device owner:

1. If you plan to install from the Play Store, add your account to your device
2. Install Video Kiosk on your device from the Play Store or by side loading the .apk file
3. Open Video Kiosk and Activate your license
4. Make Video Kiosk the home App (refer to Access Management, p.)
5. If you added an account to the device, delete the account from the device
  - Video Kiosk > Settings > Launching > Open system settings
  - Accounts > Google > [account-name] > Hamburger menu > Delete account
6. Connect your device to a computer with the Android developer tools installed
7. From a command line on the computer, enter this command:

```
adb shell dpm set-device-owner com.burningthumb.premiervideokiosk/.AdminReceiver
```

8. Optionally, add the account back onto the device
  - Video Kiosk Settings > Launching > Open System Settings
  - Accounts > Add new account > [re-add account-name]
  - Press the Back button to get back to Video Kiosk Settings

## Appendix C – Video Kiosk Settings XML File

Video Kiosk can be configured using a Settings XML file instead of modifying the setting manually on each device. This can save you a lot of time if you have a large number of devices to configure. If you update the Settings XML file on the device, Video Kiosk will reload itself using the modified settings after a short period of time.

To use a Settings XML file, use the Standard Folder Layout when installing Video Kiosk since Video Kiosk will only automatically find the Settings XML file if you place it at the following path:

```
/sdcard/videokiosk/settings/videokiosk.settings.xml
```

To use an Settings XML file to control the device settings, follow these steps:

1. **Create the File.** In a Text Editor, create a Settings XML file. For more information, refer to the Video Kiosk Settings XML Syntax Guide below.
2. **Name the File.** Ensure that you've named the file "videokiosk.settings.xml".
3. **Move the File.** Move the file onto the device either using a usb cable or over the cloud using the Cloud Download feature.

There is a **Video Kiosk Settings XML File Sample Kit available for download**. In it are some sample Settings XML project. You can use it to see what a project might look like and then use the Settings XML File in the project to make one of your own. Just copy the videokiosk.settings.xml file and modify it. The Syntax Guide is documented below. The Settings documentation can be found in the "Configuring the Settings" section of this document.

Download the sample kit from:

<https://burningthumb.com/wp-content/uploads/video-kiosk-settings-xml-file-sample-kit.zip>

### Settings XML Tags Syntax Guide

The following XML Tags are supported by Video Kiosk in the Settings XML file. The tags correspond to Video Kiosk's settings. For more information about each setting, refer to the "Configuring the Settings" section of this document.

```
<settings>
```

```
<cloud>
```

```
<bool_enabledownload> </bool_enabledownload>
```

```
<float_downloadhours></float_downloadhours>
```

```
<string_downloadlocalpath> </string_downloadlocalpath>
```

</cloud>

<fileandfolder>

<bool\_enablecache></bool\_enablecache>  
<string\_sortby></string\_sortby>  
<int\_folderdepth></int\_folderdepth>  
<bool\_enableandroidformats> </bool\_enableandroidformats>  
<bool\_enabledisplayerrors> </bool\_enabledisplayerrors>  
<string\_playlistfolder></string\_playlistfolder>

</fileandfolder>

<timers>

<int\_imagedisplayseconds></int\_imagedisplayseconds>  
<int\_imagefademilliseconds></int\_imagefademilliseconds>  
<int\_webfademilliseconds></int\_webfademilliseconds>  
<int\_videofademilliseconds></int\_videofademilliseconds>  
<bool\_enablefinishplayback> </bool\_enablefinishplayback>  
<float\_schedulecheckseconds></float\_schedulecheckseconds>

</timers>

<launching>

<bool\_launchonstart> </bool\_launchonstart>  
<bool\_restartonerror> </bool\_restartonerror>  
<bool\_enablerelaunch> </bool\_enablerelaunch>  
<int\_relaunchminutes></int\_relaunchminutes>  
<int\_restartdaily></int\_restartdaily>

<list\_restarthours>

<int\_restarthour></int\_restarthour>

</list\_restarthours>

</launching>

<buttons>

```
<bool_displaybuttonbaronlaunch> </bool_displaybuttonbaronlaunch>
<string_displaybuttonbarpattern></string_displaybuttonbarpattern>
<bool_enablepatterntones> </bool_enablepatterntones>
<bool_closepowerbuttonmenu> </bool_closepowerbuttonmenu>
<bool_disablevolumebuttons> </bool_disablevolumebuttons>

</buttons>

<interactiveplayback>
    <int_playbackmode></int_playbackmode>
    <bool_enablecontroller> </bool_enablecontroller>
    <bool_enableswipe></bool_enableswipe>
</interactiveplayback>

<systemui>
    <bool_killSystemUI> </bool_killSystemUI>
</systemui>

<sleeping>
    <int_onwakefromsleep></int_onwakefromsleep>
</sleeping>

<schedule>
    <string_schedulefolder> </string_schedulefolder>
    <string_calendarorschedule> </string_calendarorschedule>
    <group_battery>
        <bool_enablemonitorcharge> </bool_enablemonitorcharge>
        <bool_enablemonitorlevel> </bool_enablemonitorlevel>
        <int_minlevel></int_minlevel>
        <int_maxlevel></int_maxlevel>
    </group_battery>
    <group_motiondetection>
```

```

        <bool_enable> </bool_enable>
        <int_timeoutminutes></int_timeoutminutes>

</group_motiondetection>

<int_displaystate></int_displaystate>
<bool_showdate> </bool_showdate>
<string_datecolor></string_datecolor>
<bool_hdmicec_onetouchplay> </bool_hdmicec_onetouchplay>
<bool_hdmicec_sendstandby> </bool_hdmicec_sendstandby>

</schedule>

<location>

    <group_devicelocation>

        <bool_enablefixedlocation> </bool_enablefixedlocation>
        <string_fixedlocation></string_fixedlocation>

    </group_devicelocation>

    <string_locationfolder> </string_locationfolder>
    <string_geofencefile> </string_geofencefile>
    <string_locationcheck> </string_locationcheck>

</location>

<overlay>

    <string_overlayfolder> </string_overlayfolder>
    <string_overlayfile> </string_overlayfile>
    <int_scaletype></int_scaletype>
    <bool_enablehiding> </bool_enablehiding>
    <bool_enablehighlighthotspots> </bool_enablehighlighthotspots>
    <int_webviewtimeoutseconds></int_webviewtimeoutseconds>
    <bool_enablewebviewnavigation> </bool_enablewebviewnavigation>

</overlay>

<background>

    <string_backgroundfolder></string_backgroundfolder>
    <string_backgroundfile> </string_backgroundfile>

```

```

<int_nextbackgroundminutes></int_nextbackgroundminutes>
<group_audio>
    <bool_enable> </bool_enable>
    <string_url> </string_url>
</group_audio>
</background>

<management>
    <int_protocolversion></int_protocolversion>
    <string_deviceid></string_deviceid>
    <string_devicezone></string_devicezone>
    <group_postinterval>
        <bool_enable> </bool_enable>
        <int_minutes></int_minutes>
    </group_postinterval>
    <string_posturl></string_posturl>
    <list_postkeys>
        <string_postkey> </string_postkey>
    </list_postkeys>
    <bool_enablepushnotification> </bool_enablepushnotification>
</management>

<widgets>
    <bool_enablewidgets> </bool_enablewidgets>
    <bool_enablemonitorinternet> </bool_enablemonitorinternet>
    <bool_enableareaframe> </bool_enableareaframe>
    <string_padareaa></string_padareaa>
    <string_padareab></string_padareab>
    <string_padareac></string_padareac>
    <string_padaread></string_padaread>
    <bool_enablewidgetrows> </bool_enablewidgetrows>

```

```

<!-- a.b.c.d -->
<string_horizontalweight></string_horizontalweight>

<!-- ab.x.cd.x< -->
<string_verticalweight></string_verticalweight>

<list_areabwidgets>
    <string_settingsfile> </string_settingsfile>
</list_areabwidgets>

<list_areacwidgets>
    <string_settingsfile> </string_settingsfile>
</list_areacwidgets>

<list_areadwidgets>
    <string_settingsfile> </string_settingsfile>
</list_areadwidgets>
</widgets>

<security>
    <string_password></string_password>
</security>

<viewoptions>
    <bool_enablebestresolution> </bool_enablebestresolution>
    <int_orientation></int_orientation>
    <bool_enabletexturedvideoviews> </bool_enabletexturedvideoviews>
    <bool_enablepauseplayback> </bool_enablepauseplayback>
    <bool_enablepreloadwebviews> </bool_enablepreloadwebviews>
    <int_scaletype></int_scaletype>
    <bool_autorotate> </bool_autorotate>
    <bool_enableditherjpg> </bool_enableditherjpg>
    <bool_enableexiforientation> </bool_enableexiforientation>
    <int_maximagepixels></int_maximagepixels>
    <bool_enablemedianame> </bool_enablemedianame>

```

```
</viewoptions>
```

```
<legacy>
```

```
    <int_firstframecentiseconds></int_firstframecentiseconds>
```

```
    <bool_enablestretchvideo> </bool_enablestretchvideo>
```

```
</legacy>
```

```
</settings>
```

## XML Widgets

When you use a Settings XML file to configure Video Kiosk, you can add XML Widgets by specifying the path to Settings XML files for each widget. Each widget must use its own Settings XML file (i.e. don't share Widget Settings XML files for widgets of the same type).

You configure the XML widgets as described in Appendix D – Built-in Widget XML Settings, but also need to include the following tags that would normally have been added in the user widget activity user interface:

```
<widget> </widget>
```

```
<minutes></minutes>
```

```
<interactionkey></interactionkey>
```

```
<apporurl></apporurl>
```

```
<sortorder></sortorder>
```

The <widget> tag is used to configure which widget is being added and the following values are recognized:

**dateandtime**

**imagefolder**

**rss**

**url**

## Examples

Because configuring Video Kiosk using XML is non-trivial, we have prepared some examples that you can try and examine found here:

<https://burningthumb.com/wp-content/uploads/video-kiosk-settings-xml-file-sample-kit.zip>

## Appendix D – Built-in Widget Settings XML File

The Video Kiosk built-in widgets can be configured using Settings XML files. If you modify the Settings XML files on the device the widget will, after a short period of time, reload itself using the modified settings.

### XML Tags

The following XML Tags are supported by the built-in widgets in the Widget Settings XML file.

#### All

```
<settings></ settings >  
  <padding></padding>  
    <top></top>  
    <left></left>  
    <right></right>  
    <bottom></bottom>
```

#### URL Widget

```
<url></url>
```

#### Image Folder Widget

```
<pathimages></pathimages>  
<secondsperimage></secondsperimage>
```

#### Date and Time Widget

```
<pathimages></pathimages>  
<secondsperimage></secondsperimage>  
<colortext></colortext>  
<fontsizetime></fontsizetime>  
<fontsizedate></fontdatedate>  
<booleanshowtime></booleanshowtime>  
<booleanshowdate></booleanshowdate>
```

#### RSS Widget

```
<kindofwidget></kindofwidget>  
<hoursperdownload></ hoursperdownload >  
<minutesperfeed></ minutesperfeed >  
<itemsperfeed></ itemsperfeed >  
<secondsperitem></ secondsperitem >  
<fontsizetitle></ fontsizetitle >  
<fontsizedescription></ fontdatedescription >  
<colortext></ colortext >  
<colorbackground></ colorbackground >
```

```

<texthidden></ texthidden >
<itemsperfeed></ itemsperfeed >
<justifytitle></ justifytitle >
<justifydescription></ justifydescription >
<pathimage1></ pathimage1>
<pathimage2></ pathimage2>
<feeds></ feeds >
    <urlfeed></ urlfeed >

```

## Examples

The following are examples of Settings XML files for each built-in widget.

### URL Widget Settings XML File Example

```

<?xml version="1.0" encoding="UTF-8"?>
<settings>
    <padding>
        <top>13</top>
        <left>13</left>
        <bottom>13</bottom>
        <right>13</right>
    </padding>
    <url>file:///sdcard/html/analog-clock/index.html</url>
</settings>

```

### Image Folder Widget Settings XML File Example

```

<?xml version="1.0" encoding="UTF-8"?>
<settings>
    <padding>
        <top>13</top>
        <left>13</left>
        <bottom>13</bottom>
        <right>13</right>
    </padding>
    <pathimages>/sdcard/videokiosk/widget/image/images</pathimages>
    <secondsperimage>30</secondsperimage>
</settings>

```

### Date and Time Widget Settings XML File Example

```

<?xml version="1.0" encoding="UTF-8"?>
<settings>
    <padding>
        <top>13</top>
        <left>13</left>
        <bottom>13</bottom>
        <right>13</right>
    </padding>
    <pathimages>/sdcard/videokiosk/widget/date/images</pathimages>

```

```
<secondspertime>30</secondspertime>
<colortext>#FFAAAAAA</colortext>
<fontsizetime>30</fontsizetime>
<fontsize>20</fontsize>
<booleanshowtime>true</booleanshowtime>
<booleanshowdate>true</booleanshowdate>
</settings>
```

## RSS Widget Settings XML File Example

```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
  <padding>
    <top>13</top>
    <left>13</left>
    <bottom>13</bottom>
    <right>13</right>
  </padding>
  <kindofwidget>wide</kindofwidget>
  <hoursperdownload>1</hoursperdownload>
  <minutesperfeed>5</minutesperfeed>
  <itemsperfeed>10</itemsperfeed>
  <secondsperitem>30</secondsperitem>
  <fontsizetitle>20</fontsizetitle>
  <fontsizedescription>15</fontsizedescription>
  <colortext>#FF000000</colortext>
  <colorbackground>#FFFFFF</colorbackground>
  <texthidden>hidden</texthidden>
  <justifytitle>left</justifytitle>
  <justifydescription>full</justifydescription>
  <pathimage1>/sdcard/videokiosk/widget/rss/background/image1.png</pathimage1>
  <pathimage2>/sdcard/videokiosk/widget/rss/background/image2.png</pathimage2>
  <feeds>
    <urlfeed>http://rss.cbc.ca/lineup/topstories.xml</urlfeed>
    <urlfeed>http://premierwestmma.com/feed/</urlfeed>
    <urlfeed>http://rss.cbc.ca/lineup/sports-nfl.xml</urlfeed>
  </feeds>
</settings>
```

## Appendix E – Registering uncertified devices

To facilitate registering devices with Google, Video Kiosk displays the Google Services Android ID (GSAID) on a button on Video Kiosk's License Activation Activity (Screen) that links directly to the Google device registration page [Services ID button].

The steps to register an uncertified device, using the Odroid-N2 as an example, are as follows:

1. If the Google Apps are not installed on your device, download the correct Open GAPPs (for example with the Odroid-N2 board you would use ARM 9.0 Nano) from **<https://opengapps.org/>**
2. Rename the download to a short name, for example **oga.zip**
3. Sideload and open Video Kiosk
4. Allow all Video Kiosk Permissions
5. If you are not using a wired network, setup WIFI
6. Enable USB debugging
  - a. Go to the device settings (back one screen using back arrow in top left corner)
  - b. Choose About tablet
  - c. Tap Build number 8 times
7. Connect a USB cable to the device, usually using the OTG port
8. Enter these commands
  - a. **adb root**
  - b. **adb push oga.zip /cache/recovery/oga.zip**
  - c. **adb shell**
    - i. **echo "--update\_package=/cache/recovery/oga.zip" > /cache/recovery/command**
    - ii. **dpm set-device-owner com.burningthumb.premiervideokiosk/.AdminReceiver**
    - iii. **reboot recovery**
9. Open Video Kiosk
10. Tap Account
11. Tap Google
12. Wait for the "Device is not play protect certified" message
13. Enter the command **adb reboot**
14. Register Google Services Framework Android ID
  - a. Make a note of the GSAID value displayed on the Services ID button title
  - b. Tap the button to open the Registration web page
  - c. Login to Google
  - d. Add the ID from step (a)
15. Enter the command **adb reboot**

## Appendix F – Basic How Tos

Here is some information on How to perform some basic Android operations.

### Setting the Android device Time Zone

Your device allows you to set the Time Zone in the Device Settings. You may also have the option to use the network provided Time Zone.

To Change the device Time Zone manually:

1. From the Home screen, touch **Apps > Settings**.
2. Touch **Date and time**.
3. Touch **Select time zone** and select the device's time zone from the list

To have the device use the network provided Time Zone:

1. From the Home screen, touch **Apps > Settings**.
2. Touch **Date and time**.
3. Check the **Automatic time zone** check box to use the network provided time zone. This option requires that the device be connected to the network.

### Copying a folder from your computer to an Android device

You can copy a folder from your computer to your Android device (Phone, tablet, Android TV) using Google's Android File Transfer App and your USB cable.

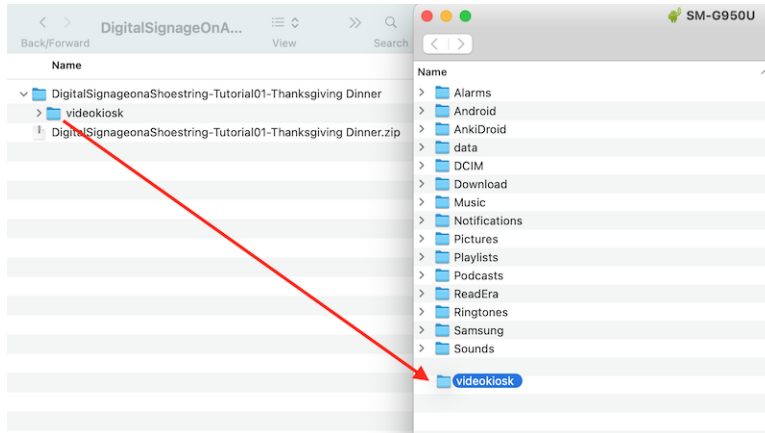
#### Requirements:

- Computer
- Android Device (phone, tablet, Android TV, Android TV Box)
- USB Cable
- Android File Transfer App

#### Instructions:

1. If you haven't already done so, download and install Android File Transfer on your computer.  
<https://www.android.com/filetransfer/>
2. Open your Android device.

3. Using the USB cable for your device, connect your Android device to your computer. Android File Transfer App should launch automatically.
4. Your Android device will display a permission dialog. Click “Allow”. The Android File Transfer Window will be displayed on your computer
5. Browse your computer for the folder you want to copy over. Drag it to the Android File Transfer Window.



## Notes:

- Some USB cables do not support file transfer and so this method might not work for you.

## Finding the Google account for your Android Device

For tutorial, go to:

<https://burningthumb.com/contact-us/app-how-to-guides-and-tutorials/android-basics-how-to-find-the-google-drive-account-for-your-android-device-phone-tablet-tv/>

## Appendix G – WebDAV

WebDAV (**Web Distributed Authoring and Versioning**) is an industry standard that you can use to host files on any web server. A large number of cloud services support WebDAV and you can create your own WebDAV server if you own a web site.

For example, cpanel users can create a Web Disk which is accessible using WebDAV.

Your web hosting provider can help you setup a WebDAV server and Video Kiosk can be configured to login to that server and download your files to all of your devices.

If you're planning on using WebDAV, we recommend you create an account on your WebDAV server specifically for Video Kiosk, and if possible, configure it in read-only mode.

You can find complete information on configuring WebDAV using cpanel here:

<https://cpanel.net/blog/tips-and-tricks/how-to-configure-and-manage-webdav-web-disks-with-cpanel/>

## Appendix H – Personal Cloud

If you run your own web site, and do not have access to WebDAV, we have developed and open sourced a Personal Cloud PHP script that will allow you to host the files for Video Kiosk on your web site and deliver them to Video Kiosk using its Cloud Download feature.

### The script on Github

You can find the script, along with basic instructions on how to use it, on Github at this URL:

<https://github.com/burningthumb/personalcloud>

Using a PHP script does require an intermediate understanding of software development so if you are not familiar with scripting you may wish to engage the services of a local web developer to help you implement your Personal Cloud solution.