# Video Kiosk – Android
# User's Manual

## Burningthumb Studios

# Table of Contents

# Introduction

Turn your Android device, in minutes, into an easy to use, reliable, robust, secure video kiosk playing videos, images and URL bookmarks in a loop. Easy to use, you'll have your Android TV, tablet, or phone playing videos, images and URL bookmarks, and running unattended, reliably and robustly with minimal set up. Designed to be easy to use, right out of the box, Video Kiosk also comes with many advanced features, such as access management, remote management, flexible screen modes and supports playback scheduling, overlays, and interactive hotspots.

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 phone, table, 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 more)
- Tablets (tested with Acer, rooted Kindle Fire, Acer, generic sub $100 tablet, and more)
- Phones (tested with Android v5 and v6 phones)
- Note: To avoid the black flash problem of amlogic SoC, use rockchip based TV boxes (like i68) or enable Textured Video Views in the Video Kiosk settings. Contact us if you are having an issue with your TV box.

## Licensing

Video Kiosk is licensed on a per device basis. There are two ways to purchase a license:

- In-App purchase
- Volume purchase

## In-App purchase

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.

Licenses purchased using In-App purchase are limited to that specific device.

## Volume purchase

To purchase multiple licenses, and receive a volume discount, do the following:

1. Send an email to **burningthumb@gmail.com** that contains the following information:
    a. The number of licenses required
    b. The account (Google, Facebook, Email, etc.) that will be used to active the licenses
    c. The email address that should receive the invoice
2. An invoice will be sent from paypal.com to the email address you provided
3. The invoice is paid using any credit card
4. The licenses are added to the license server

To activate multiple licenses do the following:

1. Add the account that will be used to activate the license to the device
2. Install Video Kiosk from the Play Store (or side load the .apk file)
3. Open Video Kiosk
4. Click the button titled Activate on the License Activity screen.
    - If you want the license to automatically expire after a preset number of days have elapsed, fill in the number of days in the field titled **Active Days** prior to pressing the button titled Activate.

Licenses purchased under the volume purchase plan are floating. When a license expires or is deactivated it, Video Kiosk returns to Free Trial mode on the device and the license returns to the pool of available licenses and may be used to license Video Kiosk on a device.

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

# Installing and Configuring Video Kiosk

## Two Step Installation

Video Kiosk is designed to be a reliable, video kiosk playing media content in a loop without any configuration. To use Video Kiosk, follow these steps:

1. Create a folder in one of the following locations and put your media in the folder:
    - /sdcard/<android id>
    - /sdcard/Movies/<android id>
    - /sdcard/Video Kiosk
    - /sdcard/Movies/Video Kiosk
2. Launch Video Kiosk.
3. There is no Step 3.

Video Kiosk will now play the contents of the media folder, in a loop, unattended. You can play videos, images and URL bookmarks, or any combination of the three. For more information about supported media formats, refer to [Supported Media , p. 11].

## About Video Kiosk's default behavior

Video Kiosk is designed to be reliable, playing media content 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 about Digital Signage Mode, refer to [Digital Signage Widget Mode, p. 6]. For more information about Interactive Mode, refer to [Interactive Kiosk Mode, p. 5].

### Playback Control

Default Playback Order

If you have not configured a Playback Order, Video Kiosk will play the contents of the Media Folder in the order of either the first playlist it finds in the Media Folder or, if a playlist is not found, 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 Order, p. 14].

## Display Control

### Image Display Time

By default, Video Kiosk will display images for 30 seconds. You can increase or decrease this display time. If you set the Image Display Time to 0 seconds, Video Kiosk will display each image and pause until the screen in touched. For more information, refer to [Supported Media , p. 11].

### Web Page Display Behaviour

By default, Video Kiosk will display web pages for 30 seconds. You can increase or decrease this display time. For more information, refer to [Supported Media , p. 11].

## Device Control

### Device Control – Dimming the screen

In order to play content, the device must be turned on and, in the case of an Android TV, the correct video input must be selected. *On any device*, Video Kiosk will automatically start playing content. If a schedule is being used, Video Kiosk will play content only when scheduled. If there is no scheduled event, Video Kiosk will, by default, dim the screen.

## 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. 41].

# Working with Display Modes

Video Kiosk's playback screen supports three different modes in which to display media – Full Screen Mode, as well as two more advanced modes - Interactive Kiosk mode and Digital Signage widget mode. The features of each are described below.

## Full Screen Mode

In Full Screen Mode, Video Kiosk will play the media in a loop, displaying it using the full playback screen. You can play videos or images or URL bookmarks, or any combination of the three.

For instructions on how to set up in Full Screen Mode, refer to [Two Step Installation, p. 3]

## Interactive Kiosk Mode

With the addition of an overlay with hotspots defined, you can turn your device into an interactive touchscreen or mouse-controlled kiosk. Define each hotspot and its associated web URL or App package name. When the a 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 media loop content, Video Kiosk monitors how long the device has been displaying the hotspot web content and returns to showing your media loop content after a configurable amount of time. This ensures that an interactive device abandoned by a user while displaying web content returns to showing your media loop content. You can change this inactivity timeout in the Video Kiosk settings.

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.*

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.

## How to turn your device into an interactive kiosk

1. Create an overlay and an overlay hotspot file. For more information, refer to [Display Features – Overlays, Interactive Hotspots, Backgrounds, p. 28]
2. Test the hotspots to ensure they are behaving as expected. For more information, refer to [Display Features – Overlays, Interactive Hotspots, Backgrounds, p. 28]

# Digital Signage Widget Mode

Video Kiosk supports Digital Signage Widget Mode, which divides the playback screen into 4 areas, A, B, C and D, allowing you to run media content in area A, while you run standard Android app widgets and/or web URL based content in areas B, C, and D.

| | |
|---|---|
| A | B |
| C | D |

When Digital Signage Widget Mode is enabled, the following will be displayed:

- Area A will display your media folder content
- Areas B, C and D will display the background or the content rendered by the app widgets and/or the web URL pages of your choice

## How to Enable Digital Signage Widget Mode

To enable Digital Signage Widget Mode,

1. Go to **Video Kiosk Settings > Widgets > Enable** and check the checkbox.
   Video Kiosk will display 4 panel display, displaying content in Area A and widgets in areas B, C and D.

For more information, refer to [Digital Signage Widgets, p. 26]
For information on Settings, refer to [Configuring the Settings, p. 52]

# Additional Features Overview

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, usability and management of your Video Kiosk.

## Media Type Support

Video Kiosk supports playback of various Media Types – video, images and URL bookmarks.

| Feature | Description | Full Screen Display | Digital Signage Display | Full Screen Interactive Overlay | Digital Signage Interactive Overlay |
|---|---|---|---|---|---|
| **Video** | Supports playback of several video formats | ✔ | ✔ | ✔ | ✔ |
| **Image** | Supports display of several  image formats | ✔ | ✔ | ✔ | ✔ |
| **URL Bookmarks** | Supports .url files (web shortcuts), which is typically used to display live web content, but can be used to reference any URL on the internet or local to the device | ✔ | ✔ | ✔ | ✔ |
| **Widgets** | Widget Areas play widgets |  | ✔ |  | ✔ |
| **URL/Apps** | Interactive Touchscreen Kiosks can open remote and local URLs as well as other Android Apps |  |  | ✔ | ✔ |

## Controlling Playback

Video Kiosk provides both simple ways and more sophisticated ways of controlling Playback Order and scheduling playback.

| Feature | Description | Full Screen Display | Digital Signage Display | Full Screen Interactive Overlay | Digital Signage Interactive Overlay |
|---|---|---|---|---|---|
| **Battery State Monitoring** | Video Kiosk can monitor the battery state on the device and stop playing to conserve battery power | ✔ | ✔ | ✔ | ✔ |
| **Playback Order Control** | Control the order media is played in using a playlist, the file and folder order, a Pick One folder | ✔ | ✔ | ✔ | ✔ |
| **Playback Scheduling** | You can schedule playback using several methods<br>• the Android Calendar on the device<br>• an XML schedule file<br>• a Google Calendar | ✔ | ✔ | ✔ | ✔ |
| **Playback Location** | You can control playback using the device location | ✔ | ✔ | ✔ | ✔ |

## Controlling the Display

There are several Display features available which control how your content looks on the device.

| Feature | Description | Full Screen Display | Digital Signage Display | Full Screen Interactive Overlay | Digital Signage Interactive Overlay |
|---|---|---|---|---|---|
| **Digital Signage Widgets** | In Digital Signage Widget Mode, the playback screen is divided into areas (4 by default), allowing you to run video content in the first area and run widgets in the other areas. You can customize the 4 areas. | | ✔ | | ✔ |
| **Overlays** | Video Kiosk supports transparent overlay files that are displayed in front of the currently displayed media. This feature is often used to overlay video with a company logo in the corner of the screen. | ✔ | ✔ | ✔ | ✔ |
| **Hotspots** | Video Kiosk supports defining hotspots on an overlay. A hotspot is a rectangle somewhere on the screen with an associated URL or App. When the kiosk is being using in interactive mode and the hotspot or key is touched, the URL or App will be displayed in a web view. | | | ✔ | ✔ |
| **Image Display Time** | You can configure how long to display still images using this feature. | ✔ | ✔ | ✔ | ✔ |
| **Fade In / Fade Out of Images, Web content, Video** | You can use a fade transition between media files. Use these features to configure how long the fade takes. (Supported on Android v4.2 and later.) | ✔ | ✔ | ✔ | ✔ |
| **Screen Orientation** | You can change the screen orientation for the media playback screen. Supported orientations include the default for the device, as well as Landscape, Portrait, Reverse Landscape, Reverse Portrait. (Supported on Android v4.2 and later.) | ✔ | ✔ | ✔ | ✔ |
| **Black flash reduction** | You can use textured video views to reduce or eliminate the black flash between videos. | ✔ | ✔ | ✔ | ✔ |
| **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.

| Feature | Description | Full Screen Display | Digital Signage Display | Full Screen Interactive Overlay | Digital Signage Interactive Overlay |
|---|---|---|---|---|---|
| **HDMI CEC support** | Video Kiosk can send HDMI-CEC commands to keep the TV on and the correct input selected. (Supported on Android v5.1and 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 scroll to the next or previous media item during playback. | | | ✔ | ✔ |

## Reliability Features

Video Kiosk is designed to provide robust, reliable playback.

| Feature | Description | Full Screen Display | Digital Signage Display | Full Screen Interactive Overlay | Digital Signage Interactive Overlay |
|---|---|---|---|---|---|
| **Behaviour if video can't be played** | Video Kiosk will skip to the next video file. | ✔ | ✔ | ✔ | ✔ |
| **Behaviour if too many errors detected** | Video Kiosk will try to correct the problem by restarting. | ✔ | ✔ | ✔ | ✔ |
| **Seamless handling of content changes** | If the media folder contents change, the new selection of media will play the next time the video loop starts. This allows you to remotely update the content without having to stop and re-start the device. | ✔ | ✔ | ✔ | ✔ |
| **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.

| Feature | Description | Full Screen Display | Digital Signage Display | Full Screen Interactive Overlay | Digital Signage Interactive Overlay |
|---|---|---|---|---|---|
| **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. | ✔ | ✔ | ✔ | ✔ |
| **Kill system UI** | This feature hides the system UI for a rooted | ✔ | ✔ | ✔ | ✔ |

| Feature | Description | Full Screen Display | Digital Signage Display | Full Screen Interactive Overlay | Digital Signage Interactive Overlay |
|---|---|---|---|---|---|
|  | device. |  |  |  |  |

## Remote Management and Update

Video Kiosk was designed to be remotely manageable so that both the media that will be played and Video Kiosk functionality can be controlled from a central location

| Feature | Description | Full Screen Display | Digital Signage Display | Full Screen Interactive Overlay | Digital Signage Interactive Overlay |
|---|---|---|---|---|---|
| **Remotely manage media files** | Using a Cloud Service, you can remotely change the media that will be played on Video Kiosk. | ✔ | ✔ | ✔ | ✔ |
| **Remotely Control Playback** | You can remotely change the playback order and update the schedule | ✔ | ✔ | ✔ | ✔ |
| **Remotely Control Display** | Using a Cloud Service, you can remotely change an overlay and change an overlay hotspot for an interactive kiosk. | ✔ | ✔ | ✔ | ✔ |
| **Remote Access Management** | Using a Cloud Service, you can remotely change the password for a device. | ✔ | ✔ | ✔ | ✔ |
| **Remote Status Reporting** | 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 | ✔ | ✔ | ✔ | ✔ |

# Working with Media Types

## Supported Media Formats

Video Kiosk will display videos, images or URL bookmarks 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 Bookmarks to display Web Content

Video Kiosk supports .url files (web shortcuts). This feature is typically 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.

To play web content, create a .url file and put the .url files in the media folder.

- If your media folder includes .url files, the web 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 Bookmarks

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

1. In a Text Editor, create a URL file, with the extension ".url". For file contents requirements, refer to the URL file Syntax Guide, below.
2. Place the .url files in your media folder. For further information about playback, refer to [Controlling Playback Order, p. 14].

## URL file Syntax Guide

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

The following example shows the content of a .url file that may be used with Video Kiosk. This file instructs Video Kiosk to play the BurningThumb Studios home page for 3000 seconds without caching the web page content:

```
[InternetShortcut]
URL= http://www.burningthumb.com/
TIMEOUT=3000
LAYER-TYPE=software
CACHE-MODE=no-cache
```

# Controlling Playback

## Controlling Playback Order

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

- A playlist
- The file and path order
- A .pickone folder(s)

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.

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.

### Using a Playlist

Video Kiosk recognizes playlists (m3u8 files). For more information on m3u8 files, refer to M3U - Wikipedia, The Free Encyclopedia at https://en.wikipedia.org/w/index.php?title=M3U&oldid=680180534

For information on Settings, refer to [Configuring the Settings, p. 52]

To use a Playlist, follow these steps:

1. Create a Playlist(s).
2. By default, you can put your playlist(s) in the media folder.  Optionally, you can put your playlist(s) in a different folder and configure Video Kiosk with the location of the playlist folder. Go to **Video Kiosk Settings > File and Folder > Path to Playlist folder** and set the path to the playlist folder.
3. By default, if the sort order is set to **Path or first playlist found** and there is a single playlist or a playlist with the name "playlist.m3u8" in the media folder, it will be found automatically by Video Kiosk and you do not need to select a playlist in the settings.
4. If you prefer to choose a playlist manually, pick a Playlist to play. Go to **Video Kiosk Settings > File and Folder > Sort by** and select the playlist to use from the list.

### Using the File and Folder Settings to control sort order

You can control the playback order of the contents of the media folder 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, p. 52]

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
   Random shuffle – Sorts in a random order

## Using 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 media 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.

## 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 your have identified and corrected the problem, we recommend you turn off this setting before deploying production devices.

## Sorting by Path or Filename

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

## Limiting the sub-folder depth

When controlling playback using the File and foldersSettings 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.

# Playback Scheduling

You can control playback locally using one of two methods:

- The Android Calendar on the device

- A local XML schedule file

You can control playback remotely using one of two methods:

- A Google Calendar
- A local XML schedule file that you synchronize to the device using a cloud service

To use a schedule to control playback, follow these steps:

1. Add playback events to a schedule to control playback time and to remotely manage device behavior.
2. Configure Video Kiosk with the location of the schedule. How to do this depends on which method you are using to control playback – Android Calendar, Google Calendar or by creating a local XML schedule file.
3. Choose the device behaviour when no schedule event occurs.  The default is to display a blank screen. Go to **Video Kiosk Settings > Schedule > Display Clock**  and set to on to display the clock instead. You can change the clock color using the **Video Kiosk Settings > Schedule > Clock Color** setting

Note: Whether the schedule displays content or not is controlled by the Schedule Setting **Display state**. If it is set to **Always on** the scheduled events do not turn the display off (events can be used to change the media folder that is playing(, if it is set to **On for events, otherwise off** then the scheduled event will turn the display on. If it is set to **Off for events, otherwise on**, the scheduled event will turn the display off.

For information on Settings, refer to [Configuring the Settings, p. 52]

## Using Android Calendar

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. In Android Calendar, create a calendar and add playback events using the start and end times in the Calendar to control when your content is played.
2. Optionally, you can control device behaviour by adding commands to the description field of the event. For more information, refer to the **Video Kiosk Schedule Command Reference Guide** section of this document.
3. 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.

## Using Google Calendar

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

1. In Google Calendar, create a calendar and add playback events using the start and end times in the Calendar to control when your content is played.
2. Optionally, you can control device behavior by adding commands to the description field of the event. For more information, refer to the **Video Kiosk Schedule Command Reference Guide** section of this document.
3. 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.
4. 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.

## Video Kiosk Schedule Command Reference Guide

Video Kiosk for Android may be controlled via the Android Calendar or Google Calendar. 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. You can change this behaviour by including these commands in the description of the calendar entry.

| Command | Arguments | Description |
|---|---|---|
| videokiosk_display_default | {on,off} | This command is deprecated. Please use **videokiosk_display_state** instead.<br>If you set this to **on**, the **Display state** will be set to **On for events, otherwise off**. If you set this to **off**, the **Display state** will be set to **Off for events, otherwise on**. Changing this will update the settings. |
| videokiosk_display_state | {alwayson,<br>eventon,<br>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.<br>If the **videokiosk_display** command is present it over rides this setting for that specific event. |
| videokiosk_path | /path/to/folder | Set the playback path used by Video Kiosk. This is the same as changing the playback folder in the App by clicking the folder icon in the button bar. Changing this will update the settings. |

| Command | Arguments | Description |
|---|---|---|
| | | 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 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 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 content 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

## Local XML schedule file

Video Kiosk will execute commands in a local XML schedule file. To use a local XML schedule file to schedule playback, follow these steps:

1. In a Text Editor, create an XML schedule file and use it to control playback. For more information, refer to the **Video Kiosk Schedule Syntax Reference Guide** below
   Download a sample Schedule file at
   (**http://burningthumb.com/download/videokiosk/schedules.zip)**
2. Configure Video Kiosk with the location of the schedule. You can configure the location of the Schedule that you want in two ways, by putting the schedule file in the default location or by setting the path to the schedule file in the App Settings.

   **Use the Default Location**

   - To use the default location, create a schedule file called "VideoKiosk.schedule" and put the schedule file in the same folder as your content files. Video Kiosk will automatically use this as the Schedule File.

   **Configure the Path in the App Settings**

   - Go to
     **Video Kiosk Settings > Schedule > Path to schedule folder**
     and set the path to the schedule folder.
   - Go to
     **Video Kiosk Settings > Schedule > Use a calendar of schedule**
     and select the schedule to use.

## Video Kiosk .schedule file Syntax Guide

| | | | | |
|---|---|---|---|---|
| <events><br></events> | | | A list of all playback events | Required |
| | <event><br></event> | | A single playback event | Required for event |
| | | <week><br></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><br></dayofweek> | Event occurs for a single day<br>**Valid Values:**<br>Sun<br>Mon<br>Tue<br>Wed<br>Thu<br>Fri<br>Sat | Use either <week> or <dayofweek> for each event, not both |
| | | <starttime><br></starttime> | The time the event starts (24hour clock, format "99:99") | Required for event |
| | | <endtime><br></endtime> | The time the event ends (24hour clock, format "99:99") | Required for event |
| | | <playlist><br></playlist> | A playlist to use for this event | Optional |
| | | <path><br></path> | A path to use for this event | Optional |

## Video Kiosk .schedule 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,<br>
Mon<br>
Tue<br>
Wed<br>
Thu<br>
Fri<br>
Sat

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

Here is an example of an XML file that contains two events, the first displaying the morning playlist on Monday mornings from 0600 to 1159 and the second displaying the afternoon playlist Thursday afternoons from 1200 to 2059:

```
<?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>20:59</endtime>
        <playlist>afternoon.m3u8</playlist>
    </event>
</events>
```

# Playback Location

You can control playback using locations defined by Geofences. Geofences can be specified in two ways:

- Video Kiosk .location XML files

- GeoJSON with Video Kiosk Extensions

*Its important to consider overlapping Geofences, in the case where Geofences overlap, and a device is located in the overlappling 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. In a Text Editor, create an XML location file(s) and use them to control playback. For more information, refer to the **Video Kiosk Location Syntax Reference Guide** below
2. Select Video Kiosk location file that defines the Geofences you want to use. You can select the locations that you want in two ways, by putting the location file in the default media folder or by setting the path to the location files in the App Settings

   **Configure the Path in the App Settings**

   - Go to
     **Video Kiosk Settings > Location > Path to location folder**
     and set the path to the location folder.
   - Go to
     **Video Kiosk Settings > Location > Use Geofence locations**
     and select the location file to use.

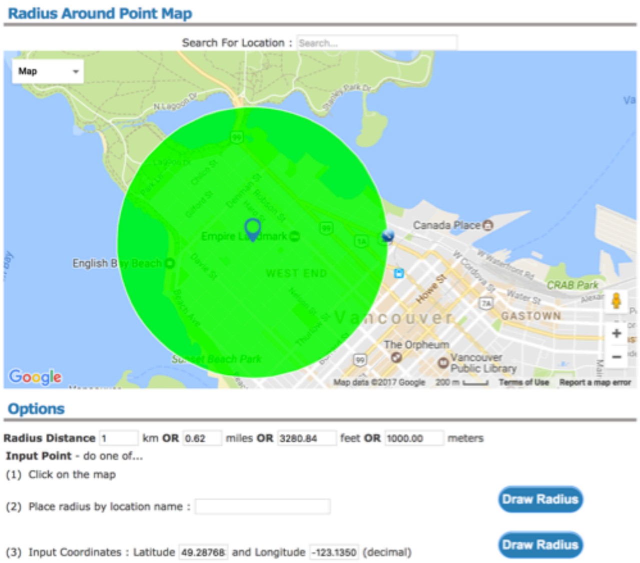
### Video Kiosk .location file Syntax Guide

| | | | |
|---|---|---|---|
| `<fences>`<br>`</fences>` | | A list of all Geofences | Required |
| `<fence>`<br>`</fence>` | | A single Geofence | Required |
| | `<name>`<br>`</name>` | The name of the Geofence | Required |
| | `<circle>`<br>`</circle>` | A circular Geofence. Defines a Geofence with a center and a radius. *Note you should select a circle OR a polygon but not both.* | Optional |
| | `<lng>`<br>`</lng>` | The longitude of the center point of the circular Geofence | Required for circle |
| | `<lat>`<br>`</lat>` | The latitude of the center point of the circular Geofence | Required for circle |
| | `<radius`<br>`</radius>` | The radius, in meters, of the circular Geofence | Required for circle |
| | `<polygon>`<br>`</polygon>` | Defines a Geofences as a series of points. Each polygon must have a minimum of 3 points. . *Note you should select a circle OR a polygon but not both.* | Optional |
| | `<point>`<br>`</point>` | One point in a polygon | |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  | <lng><br></lng> | The longitude of the a point of the polygon Geofence | Required for point |
|  |  | <lat><br></lat> | The latitude of a point of the polygon Geofence | Required for point |
|  | <playlist><br></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.



Radius Around Point Map

Search For Location : Search...

Options

Radius Distance 1 km OR 0.62 miles OR 3280.84 feet OR 1000.00 meters

Input Point - do one of...

(1) Click on the map

(2) Place radius by location name :

(3) Input Coordinates : Latitude 49.28768 and Longitude -123.1350 (decimal)

```
<?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>
```

## 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. On the web site geojson.io create the map that you wish to use.
2. Add the extra, required, fields **name** and **playlist** to each polygon on your map
3. Save the file(s) as GeoJSON (.geojson) and use them to control playback. For more information, refer to the **Video Kiosk GeoJSON Syntax Reference Guide** below
4. Select Video Kiosk location file that defines the Geofences you want to use. You can select the locations that you want in two ways, by putting the location file in the default media folder or by setting the path to the location files in the App Settings

### Configure the Path in the App Settings

- Go to
  **Video Kiosk Settings > Location > Path to location folder**
  and set the path to the location folder.
- Go to
  **Video Kiosk Settings > Location > Use Geofence locations**
  and select the location file to use.

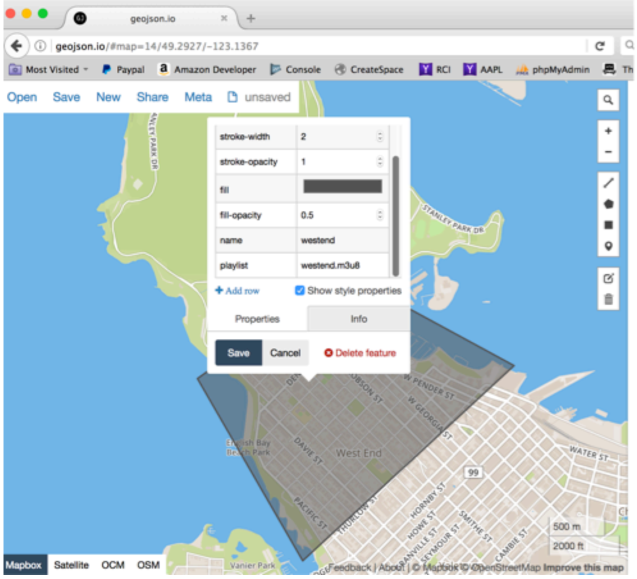
## Video Kiosk .location file Syntax Guide

Since JSON is not as human readable as XML it is stongly recommended that you use the web site 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 with the extra fields added.



| | |
|---|---|
| stroke-width | 2 |
| stroke-opacity | 1 |
| fill | ■ |
| fill-opacity | 0.5 |
| name | westend |
| playlist | westend.m3u8 |

☑ Show style properties

```
{
  "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
            ]
          ]
        ]
      }
    }
  ]
}
```

# Monitoring Battery State

With devices that have a battery, Video Kiosk can monitor the battery charging state and battery level and dim the screen if the battery is not being charged or if the battery level falls below a specific threshold.

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.

# Controlling the Display

Display features control how your content looks on the device. Digital Signage Widgets features are used to configure and customize the screen when using Digital Signage Widget Mode. Overlays and backgrounds are used to display backgrounds behind media files and overlays in front. 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, p. 52]

## Digital Signage Widgets

Video Kiosk supports Digital Signage Widget Mode, which divides the playback screen into 4 areas, A, B, C and D, allowing you to run video content in area A, while you run widgets in area B, C, D.

| | |
|---|---|
| A | B |
| C | D |

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

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

### Configuring the Digital Signage Display

To enable Digital Signage Widget Mode,

1. If you haven't already done so, enable Widget Mode.
   Go to **Video Kiosk Settings > Widgets > Enable** and check the checkbox.
   Video Kiosk will re-launch and display 4 panel display, displaying content in Area A and widgets in areas B, C and D.
2. Configure Video Kiosk  Areas B, C and D with which widgets | URLs to display.
   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 and/or web URL rendered content.  The web URL can be any valid protocol including http://, https:// (remote content) and file:// (local content).

Add any other widgets and/or web URLs to be displayed in the Area x.
Optionally, enter the amount of time, if it is different than the default value, to display it in minutes.

3. Optionally, configure the "Monitor Internet" feature. If your widgets 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, adjust the widget area layout and the widget area padding. For more information, refer to the "Customizing Widget Areas" 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 display full screen.

## Customizing Widget Areas

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 Layout

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.

You can adjust the widget layouts using **Video Kiosk Settings > Widgets > Horizontal weights** and **Video Kiosk Settings > Widgets > Vertical weights**.

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.

### Adjusting Widget Area Padding

Each widget area is padded 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.

Padding is the area between the Widget area boundaries and the sides of the Widget.

## Widget Considerations

### Where to find widgets

- Burningthumb Studios has developed some widgets specifically for Video Kiosk Digital Signage. For more information, go to http://burningthumb.com/apps/video-kiosk/video-kiosk-add-ons/
- Widgets are available from the Google Play Store.
- You can develop your own widgets and side load them onto the device.

### 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 display size (TV or tablet or phone) and how far away the viewer will be from the screen.

# Display Features – Overlays, Interactive Hotspots, Backgrounds

Display features control how your media files look on the device.

## Overlays and Overlays with Hotspots

Video Kiosk supports an overlay file with transparency (.png format with a file name that end in .overlay.png), which is displayed in front of the currently displayed media, with or without hotspots. The Overlay without hotspots feature is often used to overlay video with a company logo in the corner of the screen.

To use an Overlay, follow these steps:

1. **Create an overlay 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 content you want to use to overlay the video.  Remember to end the file name in .overlay.png so that Video Kiosk recognizes it as an overlay and not as a image to display as part of the media loop.

2. Configure Video Kiosk with the location of the overlay file. Go to **Video Kiosk Settings > Overlay > Path to Overlay folder** and set the path to the overlay folder.
3. Go to **Video Kiosk Settings > Overlay > Overlay file** and select the overlay to use.

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

## Overlay Hotspots

Video Kiosk supports a hotspot file (.xml format with a file name that ends in .overlay.png.xml), with hotspots and/or key codes defined.

### *Hotspots*

A hotspot is a rectangle somewhere on the screen and its associated URL | App. When the hotspot 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. (The URL can point to a file on the device or a file on the Internet).

If you are using Video Kiosk in digital signage 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.

### *Key Codes*

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.

### *Interaction Tracking*

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

If you are using Video Kiosk in digital signage 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.

To use hotspots, follow these steps:

1. If you haven't already done so, create an .overlay.png file. (see above)
2. Create a hotspot XML file. Using a Text Editor, create an XML file with the same name as your overlay file with .xml appended to the name. For example, if your overlay file is called "videokiosk.overlay.png" name your hotspot overlay XML file, "videokiosk.overlay.png.xml"

3. For the XML file syntax, refer to the Hotspot .xml file Syntax Guide, below.
4. Put the hotspot file in the same folder as the Overlay file so that Video Kiosk will find it automatically.
5. 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).

   Note: *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.*

6. If you are using Video Kiosk in Digital Signage 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.*

## *Hotspot .xml file Syntax Guide*

| | | | |
|---|---|---|---|
| `<hotspots>`<br>`</hotspots>` | | | A list of hotspots |
| | `<hotspot>`<br>`</hotspot>` | | 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. |
| | | `<url>`<br>`</url>` | Destination web URL used for a single hotspot |
| | | `<app>`<br>`</app>` | Destination App name used for a single hotspot |
| | | `<widgetarea>`<br>`</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>`<br>`</keycode>` | The Android key code, used to activate a hotspot using a keyboard |
| | | `<rect>`<br>`</rect>` | The screen rectangle for the hotspot defined using numbers between 0 and 1 |

|  |  |  |  |
|---|---|---|---|
|  |  |  | as a fraction of the screen |
|  |  | <top><br></top> | Rectangle top |
|  |  | <left><br></left> | Rectangle left |
|  |  | <bottom><br></bottom> | Rectangle bottom |
|  |  | <right><br></right> | Rectangle right |
|  | <intereactionkey><br></intereactionkey> |  | (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. |

### *Hotspot .xml file example*

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 .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

### Resources:

Go to http://www.burningthumb.com/apps/video-kiosk/ to download sample Overlay files or to view Video Kiosk Tutorial Videos.

## Backgrounds

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

To use a single background, follow these steps:

1. Create a background file in png or jpg format. We recommend you make it the same size as your screen (e.g. 1280×720 in size).
2. Configure Video Kiosk with the location of the background file. Go to **Video Kiosk Settings > Background > Path to Background folder** and set the path to the background folder.
3. Go to **Video Kiosk Settings > Background > Background file** and select the background file to use.

To cycle through a number of backgrounds, follow these steps:

1. Create the background files in png or jpg format. We recommend you make it the same size as your screen (e.g. 1280×720 in size).
2. Configure Video Kiosk with the location of the background files. Go to **Video Kiosk Settings > Background > Path to Background folder** and set the path to the background folder.
3. 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.

# Display Timers – Image Display Time, Fade Time, Screen

Video Kiosk includes features to allow you to customize image display time, configure and customize fade time between media files

## Image Display Time

If your media folder includes still images, each one will be displayed for 30 seconds. To increase or decrease this display time, go to **Video Kiosk Settings > Timers > Image display time** and change the display time (in seconds).

If you set the time in seconds 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.

## Image Fade Time (Android v4.2 and 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).

## Web Fade Time (Android v4.2 and 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).

## Video Fade Time (Android v4.2 and 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.*

# View Options (Android v4.2 and later)

Video Kiosk includes some features that manage specific aspects of the media playback screen.

## 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:

- Default
- Landscape
- Portrait
- Reverse Landscape
- Reverse Portrait

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

## Using textured video views 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.

# 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

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.

# Controlling the Device

For information on Settings, refer to [Configuring the Settings, p. 52]

## 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. 63]

*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

# Reliability Features

Video Kiosk is robust. If a video can't be played, due to an error, Video Kiosk will skip to the next video 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

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

For information on Settings, refer to [Configuring the Settings, p. 52]

## Behaviour 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

## 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, p. 52]

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.

*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

In OS X, you use the Finder to manage file and launch applications; in Windows, you use Windows Explorer. The Android OS equivalent is Launcher, but unlike OS X 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 called {someword}.password and put the password file in the same folder as your content files. Video Kiosk will then use “someword” as the password.
     e.g. A file called “dragon.password” is upload to the media folder. Video Kiosk will then use “dragon” as 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. Refer to [Settings - Buttons, p. 54]
   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. . Refer to [Settings - Buttons, p. 54]
   Go to **Video Kiosk Settings > Buttons > Disable volume buttons** and check the checkbox.
3. **Prevent access to Sytem 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. Refer to [Settings - , p. 55]
   Go to **Video Kiosk Settings > Rooted Devices > Kill system UI** and check the checkbox. When you exit Video Kiosk the System UI is restored.

Now, only users with the Video Kiosk password can access the device controls. If, subsequently, you want to change the volume or use any of the other device controls, uncheck the checkbox in the Video Kiosk settings, make your change and then check the checkboxes again before deploying the device again as a kiosk. Because you can only hide the system UI on a rooted device, we recommend that only rooted devices be used in any installation where users have physical access to the device.

## Advanced Access Management Set Up

Once 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 and Update

Video Kiosk was designed to be remotely manageable so that both the media that will be played and Video Kiosk functionality can be controlled from a central location. Using the Internet, you can remotely:

- change the media that will be played
- change the order in which the media will be played
- change a device's playback schedule, either by uploading a new local XML schedule file or by using Google Calendar
- change a device's Video Kiosk password
- change the overlay displayed on a kiosk
- change an interactive kiosk's hotspots
- have the device report on its status

This allows you to manage many of the Video Kiosk features without having to deploy a technician to the field.

- For more information on Playback Scheduling using Google Calendar, refer to [Playback Scheduling, p. 16]
- For more information on Device Status Reporting, refer to [Remote Device Status , p. 46]

## Remote Device Updating

You can remotely manage the media files, the playback schedule, overlays and password for a device or fleet of devices.

### How to remotely update Video Kiosk media files

Using a Cloud Service, you can remotely change the media that will be played on Video Kiosk. To do this, follow these steps:

1. Create a media folder for Video Kiosk on your device(s)
2. Set up a Cloud Service such as Google Drive or Dropbox.
3. Create a media folder for Video Kiosk on your Cloud Service.
4. Sync the media folder on the device with the media folder on your Cloud Service. If you are synching more than one device to this media folder, sync each device to your Cloud Service.
5. Move your content files into the media folder on your Cloud Service. The new media will be downloaded to all the devices that are synched to that media folder.

For more information about controlling content playback order, refer to [Controlling Playback Order, p. 14]

## How to remotely manage Video Kiosk schedule, overlays, password files

Using a Cloud Service, you can remotely change a device's Video Kiosk password, the overlay displayed on a kiosk, an interactive kiosk's hotspots and device's playback schedule, if you are controlling the schedule using a local schedule XML file. Follow these steps:

1. If you haven't already done so, create a media folder for Video Kiosk on your device(s), set up and sync a Cloud service to the device(s), following the instructions above.
2. On your Cloud Service, move any of the following files to the root of the media folder:
   - the password file
   - the overlay file
   - the overlay hotspot file
   - local XML schedule file
3. The new files will be downloaded to all the devices that are synched to that media folder.
   - For more information about creating a password file, refer to [Basic Secure Access Management Set Up, p. 43]
   - For more information about creating an overlay file or an overlay hotspot file, refer to [Display Features – Overlays, Interactive Hotspots, Backgrounds, p. 28]
   - For more information about creating a local schedule XML file, refer to [Playback Scheduling, p. 16]

# Remote Device Status Reporting

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:

1. based on a time interval
2. based on a specific time of day
3. any time using the Send Management Report button on the button bar

For information on Settings, refer to [Configuring the Settings, p. 52]

## 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:
   - readable 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.

## HTTP vs HTTPS considerations

If you plan to send the Video Kiosk Device password as a means of authenticating the post, you should use HTTPS posts instead of HTTP because it is not a good idea to send password data in the clear. 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.

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

The following keys will always be included in the POST:

androidid

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

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}

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.

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

key=value\n
key=value\n
…
key=value\n

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

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 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
setdate=YYYY:MM:DD:HH:MM:SS
```

Its important to note that 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.

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.

Setting the date and time on a device requires either that Video Kiosk is in /system/priv-app or that the device is rooted. It is important to note that 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.

## 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']));

         $l_format = 'Y-m-d H:i:s P';
         $l_date = date_create_from_format($l_format, $date);
```

```
    $l_serverDate = date_create();

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

    $l_clientTS = date_timestamp_get($l_date);
    $l_serverTS = date_timestamp_get($l_serverDate);
    $l_interval = abs($l_clientTS - $l_serverTS);

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

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

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

?>
```

# Configuring the Settings

## 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. Set this if you are using playlists. The default playlist folder is the path to the media folder.

## Settings - Timers

*Image display time*

How long (seconds) image will be displayed. Default is 30 seconds. When this is set to the value "0", Video Kisok 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.

## 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 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** |
|---|---|---|
| *Do not restart* | Does not restart daily | Does not restart daily |
| *Exit and relaunch the App* | Exits and relaunches Video Kiosk | Exits and relaunches Video Kiosk |
| *Reboot the device (requires root)* | *Video Kiosk is the Home App:*<br>Reboot Android and relaunches Video Kiosk<br>*Video Kiosk is not the Home App:*<br>Exits and relaunches Video Kiosk | 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.

*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.

### *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 in “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 a 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, you must 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, below is an example of how to lock down an Android v5+ device using the adb developer tool.

Example, locking down an Android v5+ device using the abd developer tool.

1. Add your account to the device
2. Install Video Kiosk from the Play Store or by side loading the .apk file
3. Open Video Kiosk and Activate you license
4. Make Video Kiosk the home App
5. 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 developer tools installed
7. From a command line on the computer, enter these commands:

```
adb shell dpm set-device-owner com.burningthumb.premiervideokiosk/.AdminReceiver
```

8. Optionally, add the account back onto the device
   - Accounts > Add new account > [re-add account-name]
9. Back button to get back to Video Kiosk Settings
10. 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. (Set this if you are controlling the schedule using a Local XML schedule file. If you are not using scheduling or are not using a Local XML schedule file, ignore this setting.)

*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.

*[Display default on]*

This setting is deprecated. Please use Display state instead.

*Display state*

Determines what happens when there is a calendar 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 scheduled events and will dim the screen otherwise. (Example: At the gym, use the schedule to turn the display on during classes.)

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

*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

*Path to location folder*

Choose the folder that contains your location files (XML, JSON)

*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)

*Overlay file*

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

*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.

## Settings - Background

*Path to background folder*

Choose the folder that contains your background file (png or jpg)

*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.

## Settings - Management

*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)
- Filesize (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)

## Settings - Widgets

*Enable Widgets*

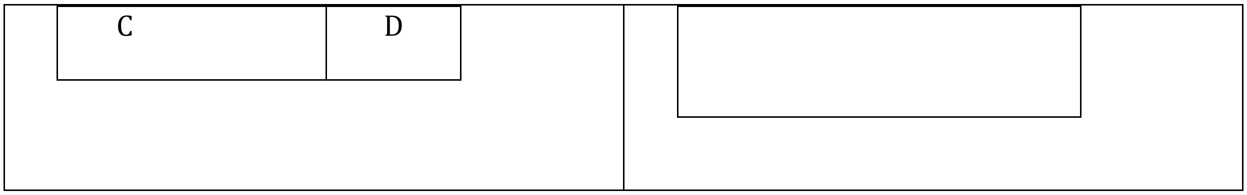
Enable widgets for this device. If you change this setting, Video Kiosk will display a 4 panel layout, including media content in Area A and widgets in areas B, C and D. If you don't configure widgets for areas B, C, D, background will be displayed. If you haven't configured a background, a black screen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

| | |
|---|---|
| A | B |
| C | D |

*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.

<table>
<tr><th>Connected to Internet</th><th>Not connected to Internet</th></tr>
<tr><td><table><tr><td>A</td><td>B</td></tr></table></td><td><table><tr><td>A</td></tr></table></td></tr>
</table>



*Draw frame on Area A*

Draw a frame around Area A

*Choose Area B Widget*

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

*Choose Area C Widget*

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

*Choose Area D Widget*

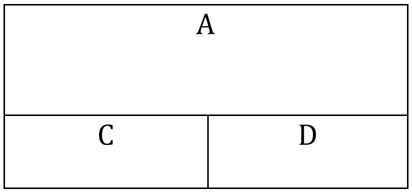
Pick and configure widgets | URLs for Area D and display time (in minutes). Default is 60 minutes.

*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.

*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. Fore 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. Fore 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:

| | |
|---|---|
| A | B |
| C | D |

## 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 (Android v4.2 and later)

*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.

*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.

## Settings - Legacy

*First frame render time*

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.

*Stretch video*

Stretch video horizontally. Typically used to stretch legacy, 960x720 video to new standard 1280x720, in order to eliminate black bands.

# 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
```