

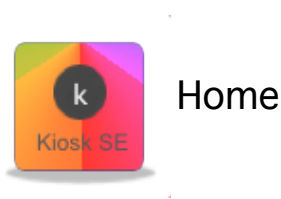
# Kiosk Browser SE

A window on your content



# Introduction

Kiosk Browser SE is a step forward from the standard editions of Kiosk Browser, incorporating useful functions and capabilities for multiple installs and remote configuration. Some of the new features in Kiosk Browser SE are:



Kiosk Browser SE is a home screen and is designed to run in place of the standard Android Launcher application. If a user presses the home key they will stay in Kiosk Browser SE. On startup users are not presented with the lock screen, just your pages. Any software failures or system restarts result in Kiosk Browser SE starting again immediately. The back button operates as a back button for the browser and does not exit Kiosk Browser SE



## Orientation

You can now lock the orientation of your devices, no amount of movement will change it.



## Sleep

You don't want your kiosk locking and going to sleep all the time, now you can switch between a number of different mechanisms for keeping your device awake and responsive. You can also put the display to sleep and wake it up at specific times



## Idle

If there's no user input you can make your kiosk jump back to its start page after a configurable number of minutes. You can also have your kiosk dim the screen while keeping your pages visible, any user interaction will cause the screen to brighten up again.



## Image Capture/Face Detection

Kiosk Browser SE can now help you fight fraud by capturing images from the screen and the front facing camera (if your device has one). Not only that, Kiosk Browser SE can, optionally, check to see if there's a face

in the picture and will only save those that will help identify individuals. An added bonus is that the space on your sdcard wont get used up too quickly.



## Image Upload

To keep your kiosk captured images safe you can upload them to your server using either ftp or sftp. Once uploaded they are deleted from the sdcard, freeing up the space. Images can be tagged to identify the kiosk they came from.



## Remote Load

When your kiosk goes idle it can also check for new configuration files, download and apply them using either http or sftp. You can update your configurations server side and be sure that your kiosks will pick them up when a user isn't interacting with them. Local files can also be updated from the remote server.



## XML Configuration

Configuration is done through preferences screens or via XML files. An initial provisioning configuration can be created and then copied to the sdcard and, optionally, deleted after being loaded. This allows you to set basic settings and later update them via the remote load/idle functionality.

# Configuration

The initial, provisioning xml file stored on the sdcard should be named **kioskse.xml**

The configuration XML is split into three main sections display, setup and config each dealing with a specific group of configuration options.

Display	Deals with features that affect the screen and how content appears.
Setup	Is the main configuration section for the kiosk covering the url, image capture etc.
Config	Deals with configuration file updates and reloading

The configuration file always begins with the xml version and the configuration content is wrapped in Kiosk Browser SE tags.

```
<?xml version="1.0"?>
  <kioskSE>
    <display>
      .
      .
    </display>
    <setup>
      .
      .
    </setup>
    <config>
      .
      .
    </config>
  </kioskSE>
```

You only need to specify the configuration sections that you are changing, adding or removing and you only need to have the elements that you want to modify in each section. For elements where you want to remove a value simply insert the appropriate open and close tags with no content in between. Once configurations are loaded from either sdcard or remote servers via http or sftp the settings are saved internally so even if your kiosk is restarted it will still have the latest settings you applied.

## Display Options

Tag	Values	Purpose
Fullscreen	true or false	If true the Kiosk Browser is displayed in full screen mode hiding the notification bar, only the application is visible*
Orientation	landscape or portrait	Fixes the orientation of the device while the application is running.
LoadingWindow	true or false	Controls the appearance of the “Loading Page...” dialogue box. True enables the feature, false disables it.
WakeLock	off, dim or bright	Controls the method of keeping the display away and the device unlocked.  off: Disables the feature and uses the system default.  dim: Causes the display to dim  bright: Puts the display at full brightness
IdleTimeout	integer representing minutes	The time after which the kiosk is considered to be idle. This can trigger configuration updates or dim the screen based on the DimOnTimeout parameter.
DimOnTimeout	true or false	Causes the display to dim if there is no user activity after the time specified in the IdleTimeout option.
DefaultZoom	integer between 1 and 250 representing a percentage of zoom.	Zooms in or out on the content to help fit it to your screen. Ideally you would use meta tags to achieve this.
EnableZoom	true or false	Allow or disallow pinch zooming of pages. True enables the feature, false disables it.
Sleep	Time using 24hr clock	The time at which the kiosk should power off the display eg 17:55 would turn off the display at 5:55 pm
Wake	Time using 24hr clock	The time at which the kiosk should power on the display eg 07:30 would turn off the display at 7:30 am
HideSystemBar	true or false	true to attempt to hide the System Bar on Honeycomb or greater devices. False to leave the System Bar alone. Requires a rooted device. Requires fullscreen to be true.

Tag	Values	Purpose
DisableLock	true or false	Disables the Android screen lock and prevents it activating. If you want to enable the lock you should set this to false. The default is to disable the lock screen.
DoNotRefresh	true or false	Prevent a page refresh when the idle timeout occurs.

\* Android tablets running versions of Android Honeycomb (3) and greater will have soft buttons visible on screen. At this time the only mechanism to remove these appears to be by rooting the device.

```

<?xml version="1.0"?>
<kioskSE>
  <display>
    <fullscreen>true</fullscreen>
    <orientation>landscape</orientation>
    <wakelock>off</wakelock>
    <loadingwindow>true</loadingwindow>
    <defaultzoom>66</defaultzoom>
    <enablezoom>false</enablezoom>
    <idletimeout>10</idletimeout>
    <dimontimeout>true</dimontimeout>
  </display>
</kioskSE>

```

Example display section

## Config Options

Tag	Values	Purpose
DeleteConfig	true or false	Specifies if the kioskse.xml file on the sd card should be deleted after it has been loaded and the configuration saved internally.
TimeoutReload	true or false	If set to true when the IdleTimeout kicks in the configuration will be loaded via http or sftp. Note the sd card file is only loaded when the application is launched.
HTTPConfig	URL to a configuration file.	Allows a configuration file to be loaded from a remote server using HTTP. Any standard web server can serve the file.
SFTPServer	FQDN of a server running sshd	Server containing xml configuration files for transfer over secure encrypted SFTP
SFTPUsername	Valid username for the SFTP server	Username used to login to SFTP server
SFTPPassword	Valid password for the SFTP server	Password to be used to login to SFTP server
SFTPFile	Remote path and filename on SFTP Server	The full path and filename of the configuration file to download from the SFTP server.

It is recommended that configuration details for the SFTP server are loaded using the provisioning kioskse.xml file which can then automatically be deleted by also setting the DeleteConfig option to true. You can still reconfigure the SFTP options from downloaded configurations if required.

```
<?xml version="1.0"?>
<kioskSE>
  <config>
    <deleteconfig>>false</deleteconfig>
    <timeoutreload>>true</timeoutreload>
    <httpconfig>http://example.com/kiosk.xml</httpconfig>
  </config>
</kioskSE>
```

Example config section using http to download configuration file

```
<?xml version="1.0"?>
<kioskSE>
  <config>
    <deleteconfig>true</deleteconfig>
    <timeoutreload>true</timeoutreload>
    <sftpserver>some.sftp.server.com</sftpserver>
    <sftpusername>ausername</sftpusername>
    <sftppassword>apassword</sftppassword>
    <sftpfile>kiosk/food_kiosk.xml</sftpfile>
  </config>
</kioskSE>
```

Example config section using sftp to download configuration file



## Setup Options

The setup section allows you to configure valid URL, upload paths and extra parameters for the kiosk. With the exception of the URL parameter, you only need to specify changes you are making from previous configurations. The url parameters MUST be specified each time, failure to do so will have the effect of disabling them and the kiosk will fall back to your `starturl` / `restrict` parameters to decide which sites and URLs are valid.

The `starturl` is a special url, any path on the same host becomes valid, if you want to restrict all hosts it's suggested that a file on the sd card is used as the `starturl` - this provides some other benefits too especially at boot and/or network disconnection or reconnection. URLs defined in the `url` section can use a wildcard (\*) to indicate "anything can come after this" this allows parameters to be passed to scripts or allows entire sites to be added to the list of valid sites. You can add multiple url lines within the configuration.

Note: If you are adding a top level site e.g. <http://kiosk.automated.it/> ensure you add the trailing '/'

```
<?xml version="1.0"?>
<kioskSE>
  <setup>
    <starturl>http://kiosk.automated.it/</starturl>
    <url>http://mycompany.com/viewproduct.php\*</url>
    <url>http://someothercompany.com/about.html</url>
    <url>http://someothercompany.com/contact.html</url>
  </setup>
</kioskSE>
```

Example setup section showing starturl and multiple url lines

If you look at the first `<url>` line you can see the url ends in a "\*" - this means that urls such as

<http://mycompany.com/viewproduct.php?productId=493495>

<http://mycompany.com/viewproduct.php?productId=493495&parameter1=fruit&lime=Y>

[http://mycompany.com/viewproduct.php\\_something](http://mycompany.com/viewproduct.php_something)

are all valid. You should match as much as possible when using the wildcard option. The seconds and third `<url>` lines will only allow access to the html files `about.html` and `contact.html` respectively.

Tag	Values	Purpose
StartURL	URL for initial site or file path	Specifies the start page for the kiosk. This is also the page Kiosk Browser SE will return to on an idle timeout.
URL	URL of a site or path	Defines a site or path that is to be considered valid and a user can visit. Multiple lines of URL parameters are allowed. None are required and if none are supplied the StartURL defines the valid path.
Restrict	true or false	Specifies if URL limiting should take place at all.
KeepCookies	true or false	If set to true then when an idle timeout refresh occurs cookies will not be cleared automatically.
ExternalBrowser	true or false	Specifies that external sites should open in the system browser.
NetworkConnect	true or false	If true the StartURL is loaded if the network disconnects and/or reconnects.
PlatformMenu	true or false	Enables or disables the ability to trigger the system configuration settings from javascript.
CaptureImage	true or false	Attempt to use the front camera to capture an image.*
OnlyFaces	true or false	Attempt to identify images with faces in them and discard those that do not.
CaptureScreen	true or false	Capture an image of the screen at the same time as an image from the front camera.
ImageTag	Unique Id	A unique name for this kiosk. Uploaded files will contain this id to identify pictures that came from a particular kiosk.
UseSFTP	true or false	Use SFTP to upload images to a server running sshd. Without this parameter images would be uploaded via FTP.
FTPServer	FQDN of FTP or SFTP server	The server to upload images to.
FTPUsername	Username	The username to login with.
FTPPassword	Password	The password to login with.
FTPDir	Path on remote server	The path on the remote FTP or SFTP server that files should be uploaded to.

Tag	Values	Purpose
FTPZip	Path and filename	Path and filename to a zip file containing resources that should be stored locally. The contents of the zip file will be extracted and stored on the sd card in the kioskse folder.
ignoreOnline	True or false	If the kiosk should ignore the fact that it may be offline. Set this to true if you only want to load local content.
MenuPassword	Values	This is an optional password to unlock the menus if they are enabled. No password would mean the menus would display. A password would mean the correct password is required to show the menus.
Storage	Values	Path to be used for storing and retrieving data. Default is the device SD Card.
DisableDownloads	true or false	Allows or prevents files from being downloaded.
CaptureVideoImage	true or false	Try and use video for capturing images. Does not work on all devices.
UploadTimer	Values	How often to try and upload images to the server. Number of minutes.

\*Requires a front facing camera.

```
<?xml version="1.0"?>
<kioskSE>
  <setup>
    <starturl>http://www.automated.it/kiosk.html</starturl>
    <url>http://kiosk.automated.it/</url>
    <url>http://blog.automated.it/</url>
    <restrict>true</restrict>
    <disablelock>>false</disablelock>
    <externalbrowser>>false</externalbrowser>
    <networkconnect>>false</networkconnect>
    <platformmenu>>true</platformmenu>
    <captureimage>>false</captureimage>
    <capturescreen>>false</capturescreen>
    <onlyfaces>>true</onlyfaces>
    <imagetag>ADR430KIOSK</imagetag>
    <useSFTP>>true</useSFTP>
    <ftpserver>myserver.somewhere.com</ftpserver>
    <ftpusername>myuser</ftpusername>
    <ftppassword>mypassword</ftppassword>
    <ftplib>/httpdocs/captures/images/</ftplib>
    <ftpzip>/kiosk/ADR430KIOSK.zip</ftpzip>
  </setup>
</kioskSE>
```

Example setup section

## Javascript Interface

Kiosk Browser SE has a javascript interface that you can use in your pages to control certain aspects and functions. These functions can be called using the Kiosk interface,

Command	Usage	Description
goBack	<code>&lt;input type="button" value="Back" onClick="Kiosk.goBack(); return false;" /&gt;</code>	Go back one page
goForward	<code>&lt;input type="button" value="Forward" onClick="Kiosk.goForward(); return false;" /&gt;</code>	Go forward one page.
reload	<code>&lt;input type="button" value="Reload" onClick="Kiosk.reload(); return false;" /&gt;</code>	Reload the page.
stop	<code>&lt;input type="button" value="Stop" onClick="Kiosk.stop(); return false;" /&gt;</code>	Stop loading the page.
endApp	<code>&lt;input type="button" value="Restart" onClick="Kiosk.endApp(); return false;" &gt;</code>	Shuts down and restarts the application.
clearCaches	<code>&lt;input type="button" value="Clear cache" onClick="Kiosk.clearCaches(); return false;" /&gt;</code>	Clear cache, history and form history.
showTitleBar	<code>&lt;input type="button" value="With title" onClick="Kiosk.showTitlebar(); return false;" &gt;</code>	Shows the title bar (disable full screen mode) - requires a restart to take effect. This gets SAVED in settings.
hideTitleBar	<code>&lt;input type="button" value="Fullscreen" onClick="Kiosk.hideTitlebar(); return false;" &gt;</code>	Hides the title bar (full screen mode) - requires a restart to take effect. This gets SAVED in settings.
showToast	<code>&lt;input type="submit" value="Android app interface" onClick="Kiosk.showToast('Hell!'); return false;" &gt;</code>	Displays a message on screen.
launch	<code>&lt;input type="button" value="Launch" onClick="Kiosk.launch('fm.last.android','fm.last.android.LastFm'); return false;" &gt;</code>	Launch another application that is installed on the device.
platformSettings	<code>&lt;input type="button" value="Settings" onClick="Kiosk.platformSettings(); return false;" /&gt;</code>	Show the platform settings screen.
enableMenus	<code>&lt;input type="button" value="Enable Menus" onClick="Kiosk.enableMenus(); return false;" /&gt;</code>	Enable the preferences and platform settings menus. This gets SAVED in settings.

Command	Usage	Description
disableMenus	<pre>&lt;input type="button" value="Disable Menus" onClick="Kiosk.disableMenus(); return false;" /&gt;</pre>	Disable the preferences and platform settings menus. This gets SAVED in settings.
screenGrab	<pre>&lt;input type="button" value="Snapshot screen" onClick="Kiosk.screenGrab(); return false;" /&gt;</pre>	Takes a snapshot of the current screen and saves it to the sd card. The file will be uploaded with and other captured images.
takePicture	<pre>&lt;input type="button" value="Take Photo" onClick="Kiosk.takePicture(); return false;" /&gt;</pre>	If possible, take a picture using the front camera. The file will be uploaded with and other captured images.
uploadImages	<pre>&lt;input type="button" value="Upload" onClick="Kiosk.uploadImages(); return false;" /&gt;</pre>	This will start an image upload process. The upload takes place in the background.
zoomIn	<pre>&lt;input type="button" value="Zoom In" onClick="Kiosk.zoomIn(); return false;" /&gt;</pre>	Zoom in on the current page. Will not function correctly for pages with fixed zoom.
zoomOut	<pre>&lt;input type="button" value="Zoom Out" onClick="Kiosk.zoomOut(); return false;" / &gt;</pre>	Zoom out of the current page. Will not function correctly for pages with fixed zoom.
wifiSignal	<pre>signal = Kiosk.wifiSignal();</pre>	Returns the wifi signal level, values of 0 (poor) to 10 (Excellent) inclusive.
wifiEssid	<pre>essid = Kiosk.wifiEssid();</pre>	Returns the essid of the wifi network the device is connected to.
batteryScale	<pre>scale = Kiosk.batteryScale();</pre>	Scale used for battery values, usually returns 100 which means x of 100, i.e. percentage.
batteryLevel	<pre>batteryValue = Kiosk.batteryLevel();</pre>	Returns the current battery level.
batteryPluggedIn	<pre>batteyState = Kiosk.batteryPluggedIn();</pre>	Returns the status of the battery/power. Will return AC, USB or Unplugged.
home	<pre>&lt;input type="button" value="Home" onClick="Kiosk.home(); return false;" /&gt;</pre>	Jump to the startURL

Command	Usage	Description
volumeUp	<code>&lt;input type="button" value="Volume Up" onClick="Kiosk.volumeUp(); return false;" /&gt;</code>	Increase the volume of the device media stream.
volumeDown	<code>&lt;input type="button" value="Volume Down" onClick="Kiosk.volumeDown(); return false;" /&gt;</code>	Decrease the volume of the device media stream.
volumeMute	<code>&lt;input type="button" value="Mute" onClick="Kiosk.volumeMute(); return false;" /&gt;</code>	Mute the volume of the device media stream
volumeUnmute	<code>&lt;input type="button" value="Unmute" onClick="Kiosk.volumeUnmute(); return false;" /&gt;</code>	Unmute the volume of the device media stream.
volumeLevel	<code>level = Kiosk.volumeLevel();</code>	Returns the media stream volume level.
volumeSet	<code>&lt;input type="button" value="Set volume 10" onClick="Kiosk.volumeSet(10); return false;" /&gt;</code>	Sets the volume of the device media stream to an absolute value.
volumeGetMax	<code>maxVol = Kiosk.volumeGetMax();</code>	Returns the maximum volume for the device media stream.
volumeMuteState	<code>volumeMuteState = Kiosk.volumeMuteState();</code>	Returns 'Muted' or 'Unmuted' depending on the mute status of the device media stream.

If you define a function "volumeChange" Kiosk Browser SE will trigger this when the volume level changes. An example function is given below.

```
function volumeChange(level) {
    alert("Volume changed to "+level);
}
```

## Additional Audio Javascript Commands

Although HTML5 audio and video are supported by Kiosk Browser SE there are some things that you just can't do with it on Android. One of those is automatically playing an audio track when a page loads. The following functions add to the already available HTML5 options. They are not extensions to HTML5 but are alternative options.

Command	Usage	Description
audioPrepare	<pre>window.addEventListener("load", Kiosk.audioPrepare('http:// www.kioskbrowser.net/somefile.mp3'), false);</pre>	Prepares the audio by filling a buffer. This is a non blocking call which will trigger a javascript function called kioskAudioReady() if it exists in your page.
audioPlay	<pre>function kioskAudioReady() {     Kiosk.audioPlay(); }</pre>	Starts playback of audio. Triggers kioskAudioPlaying()
audioStop	<pre>&lt;input type="button" value="Stop" onClick="Kiosk.audioStop(); return false;" /&gt;</pre>	Stops playback. If you call audioStop and you want to start playback again you must call audioPrepare first. Triggers kioskAudioStopped()
audioPause	<pre>&lt;input type="button" value="Pause" onClick="Kiosk.audioPause(); return false;" /&gt;</pre>	Pauses playback. Triggers kioskAudioPaused()
audioResume	<pre>&lt;input type="button" value="Resume" onClick="Kiosk.audioResume(); return false;" /&gt;</pre>	Resumes paused playback. Triggers kioskAudioResumed()
audioLoopOn	<pre>&lt;input type="button" value="Loop On" onClick="Kiosk.audioLoopOn(); return false;" /&gt;</pre>	Turns on audio looping. Triggers kioskAudioLoopState('On')
audioLoopOff	<pre>&lt;input type="button" value="Loop Off" onClick="Kiosk.audioLoopOff(); return false;" /&gt;</pre>	Turns off audio looping. Triggers kioskAudioLoopState('Off')
audioLoopState	<pre>Kiosk.audioLoopState();</pre>	Triggers kioskAudioLoopState() with a parameter equal to the loop state. On for looping, Off when not looping.



You can define functions in your javascript that will be triggered when particular audio events take place. For example:

When audioPrepare is complete and there's enough in the buffer kioskAudioReady() will be called.

```
function kioskAudioReady() {  
    Kiosk.audioPlay();  
}
```

## Accessing menus

These are multiple methods of accessing the configuration menu depending on the available device hardware or software buttons. In addition to these Kiosk Browser SE provides two gesture based methods

### 1. Fast swipe

This requires a very very fast swipe on the left hand side of the screen from the bottom all the way to the top. If your device has a lower 'System Bar' start your swipe just above it.

### 2. Multiple Tap

On some devices the fast swipe is very hard to do so in addition to that there is now the ability to use a multiple tap method:

- a) Press and hold for 2 seconds on some non clickable object on a page, eg static text.
- b) Release.
- c) Repeat a) and b) 4 more times.
- d) Single tap anywhere on the screen.
- e) Single tap again anywhere on the screen.