

# eSign Android

## **Hardware requirements**

Most Android hardware is supported. The specific requirements are:

ARM Cortex-A series CPU  
ARMv7 Instructions  
NEON Technology  
GPU  
Android OS Version  
Android API Level

<https://play.google.com/store/apps/details?id=com.ss.syscheck&hl=en>

When it shows:

- ARM CPU showing ARMv7
- OS Version listed above (or newer)
- NEON is listed as yes

Then the device should work.

## Index

Installation & Configuration.....	3
Installation Part 1 – eSign Server.....	3
Installation Part 2 – eSign Android installation and setup.....	3
General information about the eSignAndroid process.....	5
General information about the design xml.....	6
Values to read out from the doorsign xml.....	12

## Installation & Configuration

### Installation Part 1 – eSign Server

#### eSign Server

1. Install and license the eSign Server. For information see the [eSign manual](#).

2a. From [eSign Server v10](#) and on, the Server installs the template and places the design xml into the correct directory.

2b. For earlier versions do the following:

- Add the AndroidDesign.xml to the DesignXML directory of the eSign Server

(C:\esign\Files\DesignXML\AndroidDesign.xml).

- Place the design necessary files (background and images) in the eSign Server files directory

(C:\esign\Files\Android\Default). When the paths are different, make sure that the design xml refers to the correct image paths.

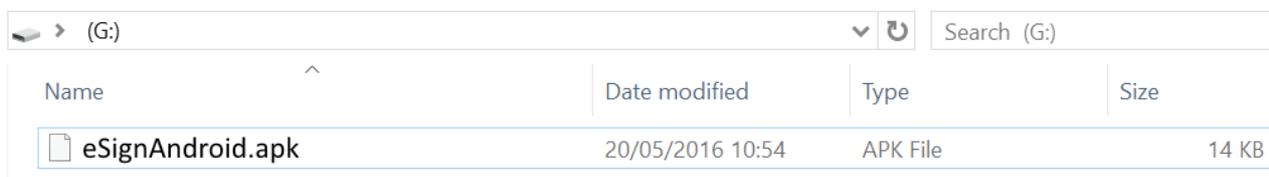
### Installation Part 2 – eSign Android installation and setup

Follow the Android device manual on how to install APK's. In general this will involve attaching a USB with the eSignAndroid.apk.

1. Start the Android device.
2. Follow the device specific installation progress, example:

-Prepare a USB drive.

-Copy the APK file onto the USB drive:



Name	Date modified	Type	Size
eSignAndroid.apk	20/05/2016 10:54	APK File	14 KB

-Turn on the device.

-Connect the USB drive with the device.

-Open settings.

-Go to storage.

-Go to USB.

-Select APK and install from there.

## Install

Path: USB Memory



Directory: USB Memory

 eSignAndroid.apk

-Else use an APK installer and navigate to the APK on the USB.

3. Start the Application and set the IP Address to match the eSign Server and note down the device ID.

4. When connecting to the eSign Server, the App will send its unique device ID to request a design xml from the server. Search in the easescreeen\_esign.ini **[EmbeddedDoorSigns]** section for the ID and set it to request the proper design xml. The default design settings would be **ID=1.1.3\_AndroidDesign\_v2\_NoPin.xml|11|300|#|3**. Save this made change and restart the Server.

5. The App now loads the design.

Additional information: the ID=1.1.3\_AndroidDesign\_v2\_NoPin.xml|11|300|#|3 stands for:

ID = the device ID.

AndroidDesign.xml = the design the resource should display (note these have to be made specifically for the Android app).

|11 = the resource ID, change this to the desire resource ID.

|300 = the pre-run time.

|# = the post-run time

|3 = the board ID.

Note: We try to keep all functionality backwards compatible, the only major difference are the changes between v1.1.2 and above. This can be seen in how the pop-ups and actions are addressed.

Designs delivered with the actual version are:

1.1.3\_AndroidDesign\_v2\_NoBook.xml

1.1.3\_AndroidDesign\_v2\_NoPin.xml

1.1.3\_AndroidDesign\_v2\_PIN.xml

## General information about the eSignAndroid process

The eSignAndroid works as followed.

### Application launch and set IP address

The application launches into a flat base design and scans for the saved IP address. If no IP address is set (first launch), the IP address can be set directly – or if the set IP address can not be reached, the Admin can login with the default user:Admin and the default password:FELDTECH and then set the IP address.

*Note: just the IP address is required, e.g. '10.0.0.10' or 'esign.easescreen.com'.*

### Requests made to the server

Upon successful connection, the application will request the download of the design xml (by default the AndroidDesign.xml). At a set interval the application will request the doorsign xml (containing the event data). At a longer set interval (60 sec), the application will request data for the overview page to update all rooms in the overview section.

### Application loop

Every loop consists of downloading the required xml(s). Parsing their contents, creating images, buttons and labels.

### User interaction

The application allows user interaction through touch. The buttons allow navigation to new pages (where new images, buttons and labels are displayed) as well as triggering actions. The following actions are available:

- ChangePage
- Book
- Cancel
- Confirm
- Extend
- Support Queries
- Increment
- Decrement
- Login
- Admin Login
- SetIP
- Exit

### Additional interaction:

A native Android touch keyboard is used for submitting IP addresses or logging in to book events. Swiping gestures are enabled for the overview page to swipe through the room list.

## General information about the design xml

The design xml contains all the information responsible for the visual representation of the eSignAndroid. This design can be customized to fit a desired (company) style.

The design xml exists of two parts: GeneralSettings and Visuals. The GeneralSettings are loaded once upon start or on design change. The Visuals are used on every loop to match the data returning from the doorsign xml.

### GeneralSettings

Found as: `<Settings><GeneralSettings>` contains the following nodes:

#### `<DesignXML>Android_Design</DesignXML>`

If the design does not contain this exact node under GeneralSettings, the application will load the default base design displaying the error of a mismatching design xml. This is, because the eSignAndroid relies on various settings in the GeneralSettings, which the regular eSign design xml designs do not contain.

`<DesignVersion>1.1.3</DesignVersion>` Allows the Application to check which functionality is can be used and which variant of older features have to be used instead.

#### `<ScreenOrientation Orientation="Landscape">`

If not set the default device ScreenOrientation is used. The allowed values depend on the device, the values that can be defined are: [Landscape](#), [Portrait](#), [InvertedLandscape](#) and [InvertedPortrait](#).

#### `<DimScreen>`

The brightness of the screen is defined here. This is done in a byte value counting from 0 to 255, where 0 is the minimum brightness and 255 is the maximum brightness.

Under the `<DimScreen>` node the specific times for the brightness is defined in the `<Dim>` node. `<Dim Start="00:00" End="08:00" Value="100"/>` Start defines the start time, End the endtime and the Value the brightness.

Note: to set a value from before midnight, to past midnight, two entries must be made as followed. `<Dim Start="22:00" End="23:59" Value="20"/>` and `<Dim Start="00:00" End="08:00" Value="20"/>`

#### `<Timer Interval="15000">`

This value defines the application loop interval in milliseconds. This influences how often the application sends the static requests to the server to update the information. To reduce stress on a server, a lower interval value can be used e.g. once a minute (60000).

`<ClockFormat>%CLOCK|dHH:MM%</ClockFormat>` Clockformat defines a datetime format for all `<Label xxx="Clock">` note: the app updates at intervals, the clock will thus not be updated every second rendering hh:mm:ss useless

## <Background>

Contains all relevant background elements. These backgrounds are downloaded from the eSign Server. It is essential for the application that these nodes are defined, even if no different background state is desired (define the same background for each case to work around).

Found under the <Background> node are:

<BGHasEvent> the background when the resource has an event.

<BGIsBookable> the background when the resource is available.

<BGUnconfirmed> the background when the resource is awaiting confirmation.

<BGPopup> the background of the popups (note size limitation less than 50% of screen height due to space availability for the keyboard).

<BGDefault> the default background, this is loaded directly from the Android device. We do not recommend this to be edited manually as it requires one to navigate to the installed application on the device itself (differs per hardware).

*Note: make sure that these pictures paths are correct!*

*Hint: try to save them as economically as possible to reduce bandwidth and memory usage (no resolutions bigger than the device resolution or high print DPI pictures which will only result in jagged images from downscaling).*

## <Pictures>

Contains the images required for the popup dialogues. These can be altered to match a corporate design or kept as is.

<Popup\_OK>URL|FILES/Android/Default/Buttons/Ok.jpg</Popup\_OK> the OK button.

<Popup\_Cancel>URL|FILES/Android/Default/Buttons/Cancel.jpg</Popup\_Cancel> the Cancel button.

## <Overview>

Contains all settings regarding the Overview design. These nodes need to be defined. If it is not desired to show the overview, simply change the OverviewPage to a page which can't be navigated to.

<BookText>Do you want to book room: %OVERVIEWLOCATION% for %BOOKDURATION% minutes?</BookText>

Defines the booking popup text.

<CancelText>Do you want to book room: %OVERVIEWLOCATION% for %BOOKDURATION% minutes?</CancelText>

Defines the overview booking popup text.

## <PopupParamters>

Defines the overview popup response text (meaning what will be displayed upon a successful OK, ABORT).

<BookPopupOK>Booking room: %OVERVIEWLOCATION%</BookPopupOK>

<BookPopupAbort>Could not book room: %OVERVIEWLOCATION %</BookPopupAbort>

**<CancelPopupOK>**Canceled event: %OVERVIEWSUMMARY  
%</CancelPopupOK>

**<CancelPopupAbort>**Could not cancel the event: %OVERVIEWSUMMARY% of  
room: %OVERVIEWLOCATION%</CancelPopupAbort>

**<OverviewPage Page="2"/>**

Defines on which Page the OverviewLayout will be shown.

**<OverviewLayout x="1|%x" y="253|py" width="82|%x" height="70|%y"/>**

Defines the positioning of the **OverviewLayout**. The OverviewLayout contains the individual rows containing specific resource (room) data.

**<RowFont name="Calibri" size="24" color="0F2151" style="Bold"/>**

Defines the font settings for each row.

**<RowPosition x="1|%x" y="1|%y" width="82|%x" height="76|py" rowcolor="b7bccca" altrowcolor="e7e8ed" strokecolor="FFFFFF"/>**

Defines the positioning of the individual rows. The rowcolor and the altrowcolor define the usual and alternative row color.

**<Items>**

**Contains which specific nodes or texts should be shown for each row. This feature is not yet operational and can be ignored in the version 1.0.0. The same goes for the underlying nodes.**

**Currently only a hardcoded LOCATION, SUMMARY, DT-START – DT-END, and optionally BOOK is shown based on the Pin=On value.**

**<Title x="1|%x" y="1|%y" width="30|%x" height="76|py" Text="%SUMMARY"/>**

**<Location x="30|%x" y="" width="52|%x" height="76|py" Text="%LOCATION"/>**

**<Logo x="30|%x" y="" width="52|%x" height="76|py" Text="%X-WR-PICTOGRAM"/>**

**</Items>**

**</Overview>**

**<Board>4</Board>**

Defines the Board ID to use for the overview.

**<Templatesize width="1280" height="800"/>**

Defines the resolution used for the design. The % values are based on this template size.

*Note: an incorrect value will lead to incorrect and possible erroneous behaviour.*

**<Booking isEnabled="True" PIN="On" Increment="10"/>**

Choose if direct booking is enabled by setting isEnabled to either True or False, to set PIN="On" or PIN="Off", and define the Incremental steps.

### **<SrcXML eSignServerUrl="SetIP" Port="28888"/>**

Sets the Server IP address and the Port to reach the eSign Server. The default value is set to "SetIP", allowing on the first start-up, the user to enter an IP address. All interactions to change the IP address at later stages will require the Admin rights to change the IP address. The port 28888 is the default port to reach the eSign Server.

### **<Users>**

Contains the login data for usermanagement.

#### **<User name="Admin" password="FELDTECH"/>**

There must always be an Admin user, make sure that the default password is changed and do not lose this password! In the case of a mistake, the design can be altered and sent to the device again as long as the device can reach the new design xml.

#### **<User name="User" password="user"/>**

Numerous users can be defined. Specify the name="xxx" and the password="xxx" to allow booking for that specific user. Note, this is design bound. Every resource using designA, can only allow the users set in designA.

### **</GeneralSettings>**

The end of the GeneralSettings section.

### Visuals

**<Visuals>** contains the actual buttons, images and labels for each page.

**<Visuals page="0">** the 0 defines which page this is. The page 0, is always loaded first and the application will automatically return to this page after user inactivity.

*Note: it is most important, when using multiple pages, that these are numbered sequentially (0,1,2,3... and not 1, 4, 6).*

### **<Buttons>**

Contains all buttons to be printed for the **<Visuals page ="X">** page.

#### **<Button ImageButton="Always">**

The node Button contains the Position, Action and Path properties defining the button. There are various button types. The assignable attributes are:

**<Button ImageButton="Always">** the button will always be displayed on this page.

**<Button ImageButton="NoEvent">** the button will be displayed when there is no event.

**<Button ImageButton="HasEvent">** the button will be displayed when there is an event.

**<Button ImageButton="HasNode%">** the button will only be displayed when the respective node can be found. For example HasNode%VEVENT/ORGANIZER, will only return true when in the webinterface the organizer has been entered for that event.

This can also be used to check for future events by using the following: **HasNode%/VEVENT[2]**.

The button will be displayed if Event 2 can be found.

**<Button ImageButton="Unconfirmed">** the button will always be displayed when an event has to be confirmed.

**<Button ImageButton="HasError">** the button will be displayed when the application encounters an error.

**<Button ImageButton="Admin">** the button will be displayed when an Admin user is logged in.

**<Button ImageButton="LoggedIn">** the button will be displayed when a user is logged in.

**<Button ImageButton="LoggedOut">** the button will be displayed when no user is logged in.

**<Button BookingNumberBox="Always">** is the only button which requires font settings. This displays the number of minutes for which the event will be booked.

**<Button IncrementButton="Always">** this button increments the number of minutes to book with 15.

**<Button DecrementButton="Always">** this button decrements the number of minutes to book with 15.

**<ActionParameters Request="Book" DSID="%DSID%" DURATION="%BOOKDURATION%" RESID="%RESID%" />**

ActionParameters define the action to be executed and with which Parameters. Each Action contain their own respective Parameters.

#### Actions

For each **Request** Actions the Parameters are listed below:

**Request="Book"** requires the Parameters: **DSID="%DSID%"**.

**DURATION="%BOOKDURATION%" RESID="%RESID%"**.

**Request="Confirm"** requires the Parameters: **DSID="%DSID%" UID="%UID%"**.

**Request="Cancel"** requires the Parameters: **DSID="%DSID%" UID="%UID%"**.

**Request="Extend"** requires the Parameters: **DURATION="%BOOKDURATION%"**.

**DSID="%VEVENT/X-WR-LOCATIONID%" UID="%VEVENT/UID%"**.

**DSID** is an identifier to the Doorsign ID (**%DSID%** points to the Unique Doorsign ID (for example seen in the EmbeddedDoorsign section of the hardware.ini) and **%VEVENT/X-WR-LOCATIONID%** points the Location ID of the currently playing event (note this value does not work when booking an event, as there is no VEVENT node available when the room is empty)).

**DURATION** stands for the number of minutes for which the room should be booked.

**%BOOKDURATION%** stands for the number displayed on the BookingNumberBox.

**RESID** stands for the Resource ID which has to undergo the respective request. Where **%RESID%** points again to the own device its resource ID.

**UID** The UID stands for the event ID, where **%VEVENT/UID%** stands for the first active event found on the resource (used for extending, confirming and cancelling of events).

**<Action Action="ChangePage" Page="0" />**

Action uses its value 'ChangePage' to change the current page to another page. The Page= value defines to which page should be changed.

**<Action Action="Increment"/> <Action Action="Decrement"/>**

Both the Increment and Decrement use the values as defined under the <GeneralSettings> node:  
<Booking xxx xxx Increment=Value/>, to either increment or decrement the  
**%BOOKINGVALUE%**.

Support query Actions require the following Parameters: DSID, RESID, X-WR-CALID, SUBJECT and BODY.

**<ActionParameters Request="EMAIL" DSID="%DSID%" RESID="%X-WR-CALID%" SUBJECT="Meetingroom Information System: Facility needed" BODY="request for support with light/shades/power"/>**

With this an email will be requested at the eSign Server and sent with the above Subject and Body to the respective intended (for more information see the eSign Server HOWTO).

To Set the Server IP address:

**Action="SetIP"**

Which opens a form with a field for the IP address and Port of the to be set server address. After clicking the OK button a connection will be initiated straight away, upon a successful connection the IP address is stored and the App will start to reload (a new Server might have a new Design for the newly added Resource).

**<Action Action="Exit"/>** Allows the user to exit the application (it is recommended that this action can only be visible/executed when the user is logged in as an Admin).

**<Action Action="Logout"/>** Allows the user to logout (every user, especially Admins, should log off after executing their user specific actions).

**<Action Action="Authenticate"/>** Allows the user to login as an Admin, allowing the Admin user to execute Admin specific designed queries (these can be defined by yourself) (Requires the **<PopupType>2</PopupType>** node under **<PopupParameters>**).

The PopupParameters define the text to be displayed when their respective conditions apply.

**<PopupParameters>**

**<PopupText>**Do you want to book for %BOOKDURATION% minutes?**</PopupText>**

The PopupText is the text which is shown on the Popup.

**<PopupOK>**Booking room: %X-WR-CALNAME%**</PopupOK>**

The PopupOK is shown when the condition 'OK' comes forward ('OK is clicked' or IP has successfully been set or the PIN has been accepted).

**<PopupAbort>**Could not book room**</PopupAbort>**

The PopupAbort is shown when the condition 'ABORT' comes forward ('CANCEL' is clicked, PIN was not correct, setting the IP has failed because it could not connect to the set IP address).

**<PopupType>2</PopupType>**

The PopupType defines what kind of popup will appear. There are 4 types of popup forms:  
PopupType 0: Contains just an OK button. This form is used to show plain informative text.  
PopupType 1: Contains an OK and Cancel button, allowing choices to execute actions.

PopupType 2:

**</PopupParameters>**

Each Button has their own **<Position>**, **<Path>** and **<Action>** child node. Which will be explained further on in the document.

**</Buttons>** end of the buttons section for the page.

**<Images>** contains all images in **<Image>** nodes for this page.

**<Image Image="HasEvent">** contains the positioning and image path for the respective image.

For the image attributes see the Buttons ImageButton="" attributes.

Each Image has their own **<Position>** and **<Path>** child node. Which will be explained further on in the document.

**</Images>** end of the Images section for the page.

**<Labels>** contains all the text labels in **<Label>** nodes for this page.

**<Label name="Attribute">** the name can be a personal identifier for the label. The attribute allows all aforementioned attributes from the Buttons as well as the Images and is extended by the following attribute:

**<Label name="Clock">** the clock attribute will show the device time with the previously set

**<ClockFormat>%Clock|dHH:mm apem%</ClockFormat** format.

**</Labels>** end of the Labels section for the page.

#### General node elements

**<Position x="0|px" y="0|py" width="200|px" height="200|py">**

**<Position x="0|%x" y="0|%x" width="200|%x" height="200|%y">**

Many nodes make use of the **<Position>** childnode. This is a necessity or the application can not render the design correctly. Make sure that all of the node's attributes have a valid value. The nodes that require the **<Position>** child node are: **<Button>**, **<Image>** and **<Label>**.

**x=""** stands for the x (horizontal) axis.

**y=""** stands for the y (vertical) axis.

**width=""** stands for the width of the to be designed object.

**height=""** stands for the height of the to be designed object.

The values are separated by a pipe (|), the identifiers behind the pipe tell the software if pixels in either x (|px) or y (|py) axis are required, or percentages in the x (|%x) or y (|%y) axis. The % values are relative to the previously set **<Templatesize width="1280" height="800"/>** node.

The values for each attribute should not exceed the resolution boundaries of the screen or it will not be displayed.

**<Path>URL|Files/Folder/imagename.extension</Path>** sets the path to reach the respective image. Make sure that the extensions are not forgotten (like .png, .jpg). The '**URL|**' part defines that the picture is located on the eSign Server.

**<Font>** defines all font properties of the respective label or numberbox.

The **<Font>** node can contain the following attributes:

**name=""** this defines the font family, these families must be present on the device. Flexible installation or usage of fonts will be introduced in next updates of the application.

**size=""** defines the font size.

**color="xxxxxx"** requires a 6 digit hex colour code (no alphavalues).

**align=""** sets the text alignment of the label (leading by default). The options are: Leading, Trailing and Center.

**style=""** defines the textstyles of the label to be: Bold, Italic or Underline.

**autosize=""** sets the label to increase its size based upon text length. Use this carefully with the texttrim and wordwrap options, the default value will be false.

**texttrim=""** defines how the text will be cut off when it reaches the labels boundaries. This can be set to "word", "none" or "character".

**Wordwrap=""** will word be wrapped together at the line break or not, the value "true" or "false" can be used , the fault value is false.

*Note: when setting the autosize, texttrim or wordwrap values, the label requires to be large enough to contain the text. Depending on the set values, the application can try to force the text in a too small size and render strange looking characters. In this case, enlarge the container (the label) or reduce the text to fit with the chosen settings.*

## Values to read out from the doorsign xml.

Values that are requested from the doorsign xml or application specific variables are defined as followed:

### Device specific

**%CRLF%** defines a carriage return line feed (enter/return).

**13#10%** the same as '**%CRLF%**'.

**%ANDROIDIP%** contains the IP address of the Android device. Can result in 0.0.0.0 or the local loopback address when multiple network interfaces are present.

**%CLOCK%|dHH:mm ampm** shows the actual system time in HH:mm ampm format.

Note: for the clock to show AMPM, depends on the device time settings.

**%DSID%** contains the unique doorsign ID. This is used in the eascreen\_esign.ini to assign the correct design xml to the device.

**%LASTREQUEST%** contains the last request made to the server, this comes in handy when there is an error.

**%LASTERROR%** contains the last known error, this comes in handy when there is an error.

**%SERVERURL%** contains the stored server url.

**%BOOKDURATION%** displays the set value for booking an event.

**%OVERVIEWSUMMARY%** contains the title of the overview event (e.g. when canceling).

**%OVERVIEWLOCATION%** contains the location of the overview resource (book for room: location?)

**%NOW%** contains the current date-time on the device, must be followed up with a pipe '|' and a

date time format.

Doorsign static values

**%X-WR-CALNAME%** contains the location name of the resource.

**%X-WR-CALNAME-X%** contains the description of the resource location.

**%X-WR-CALNAME-INFO%** contains the location description (e.g. basement, upperfloor).

**%X-WR-CALID%** contains the Internal ID of the resource as found in the esign server web interface under Administration > Resources > Details.

Doorsign event values

**%VEVENT/DESCRIPTION%** contains the events description.

**%VEVENT/SUMMARY%** contains the events title.

**%VEVENT/X-PERSONS%** contains the number of persons expected for this event.

**%VEVENT/DTSTART%** contains the start date of the event. - Use as followed:

**%VEVENT/DTSTART|dHH:mm%** with a format separated by a pipe '|'.  
Use as followed:

**%VEVENT/DTEND%** contains the end date of the event. Use as followed.

**%VEVENT/DTEND|dHH:mm%** with a format separated by a pipe '|'.  
Use as followed:

**%VEVENT/ORGANIZER%** contains the customer name of the event.

**%VEVENT/CONTACTNAME%** contains the contact name linked to this event.

**%VEVENT/X-COMMENT%** contains the comment to this Event.

**%VEVENT/X-WR-LOCATIONID%** contains the location's ID (resource ID on the website).

DateTime Formats can be:

- d** Displays the day as a number without a leading zero (1-31).
- dd** Displays the day as a number with a leading zero (01-31).
- ddd** Displays the day as an abbreviation (Sun-Sat).
- dddd** Displays the day as a full name (Sunday-Saturday).
- g** Displays the period/era as an abbreviation (Japanese and Taiwanese locales only).
- gg** Displays the period/era as a full name.
- e** Displays the year in the current period/era as a number without a leading zero (Japanese, Korean and Taiwanese locales only).
- ee** Displays the year in the current period/era as a number with a leading zero (Japanese, Korean and Taiwanese locales only).
- m** Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
- mm** Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
- mmm** Displays the month as an abbreviation (Jan-Dec).
- mmmm** Displays the month as a full name (January-December).
- yy** Displays the year as a two-digit number (00-99).
- yyyy** Displays the year as a four-digit number (0000-9999).
- h** Displays the hour without a leading zero (0-23).

- hh** Displays the hour with a leading zero (00-23).
- n** Displays the minute without a leading zero (0-59).
- nn** Displays the minute with a leading zero (00-59).
- s** Displays the second without a leading zero (0-59).
- ss** Displays the second with a leading zero (00-59).
- z** Displays the millisecond without a leading zero (0-999).
- zzz** Displays the millisecond with a leading zero (000-999).
- ampm** Uses the 12-hour clock for the preceding h or hh specifier, and displays the contents of the TimeAMString global variable for any hour before noon, and the contents of the TimePMString global variable for any hour after noon.

*Note: Incompatible changes may influence the playback or even crash the application.*

*Hint: Please make a backup of each working design.*

*Hint: When changing the design for multiple devices, test run the design onto one device to check if the results are as desired.*