# Photobook Corona SDK Project Documentation

By David Gross

Version Monday, October 16, 17

This documentation describes the inner workings of the "photobook" lua module and sub-modules, written for Corona SDK. The photobook system is designed to show visual books on tablet devices, especially the Apple iPad.

Table of Contents

Photobook Corona SDK Project Documentation 1

Design 5

Multitouch (Pinch and Zoom) 5

Object attribute: canPinchZoom 5

Zooming 5

Testing Multitouch in the Corona SimuLator 5

Photobook Wordpress Publication 6

Overview 6

General Wordpress Setup notes 6

Installation and Setup 6

Set Up Your Blog 7

Unique ID 7

Categories 7

Publishers 7

Build a book 8

Chapter Order 8

What if you have only one chapter? 8

What about sub-chapters? 8

Pages from Posts 8

Text 8

Pictures 8

Audio and Video 8

Custom Fields 9

Send A book for publication 9

Method #1: Publish from your blog 9

Method #2: Request publishing programmatically 9

Shelves 9

Corona File structure 9

iOS File Structure 9

Android File Structure 10

Custom Files 10

ISBN Numbers 10

Build Files 10

Building the App 12

Tasks to build and upload an app to the itunes App store 12

Checklist 12

Artwork 12

Library Title Bar 12

Icon, DefaultLandscape/Portrait 12

Provisioning Profiles 13

Settings 13

Build Settings file, Config File 13

Choose a short App Name 13

Facebook 13

Twitter 13

Fonts 13

Font Metrics 13

Mapping an Image 13

How to share demo apps with others 14

1) Sending Your Device ID to a Developer 14

2) Installing an Application for Testing 14

Exporting Chapters from Adobe InDesign 15

Setting up your InDesign File 15

Object Styles and Setting XML Attributes of Objects 15

Captions and Overlays 15

Hyperlinks (Actions) 16

Custom Fields 17

Buttons 17

Audio buttons 17

Page Attributes 17

Exporting Chapters from Adobe Lightroom 17

Facebook Integration: Must have picture stored on the internet 19

Important!!! 19

Writing on the Wall 20

Uploading Pictures 21

Books 21

Book Table 22

Page Layering 22

Text Formatting 23

Indenting Lines 23

Formatting Commands 23

Text Style Sheets 24

Special Characters in XML 25

Conditions 25

The Condition 25

Multiple Conditions 25

Examples 25

Condition Results 25

How to Show Multiple Languages or alternative Texts 26

The Inventory 28

Locked page example 28

In-App Purchases 29

Inventory Item Elements: 30

Introductions 31

Book XML Coding 31

Comments: 31

Book Attributes in the XML: 32

Chapter Attributes in the XML: 34

Page Attributes in the XML (See above for most of these): 36

Book Elements: 38

Chapter Elements: 38

Page Elements: 38

Plugins 69

Templates 72

PAGE METADATA UPDATE FILES 73

InDesign Bookmaker Plugin 74

How to install 75

Initial InDesign File Setup 75

Preparation of an Existing InDesign File 75

Using the Script 75

Field Names 76

Sample document 76

Developing Plugin Scripts 77

# Design

## Multitouch (Pinch and Zoom)

The app supports multitouch gestures in iOS, and possibly in Android. This means you can use two fingers for objects which support pinch, zoom, move, and rotate.

You must tag an element if you want it to respond to gestures. Only images currently can use multitouch.

In terms of design, you should be aware of how we implemented multitouch. The concept is that not everything should be available for manipulation, but it must be clear to the reader what can be controlled.

There are two build-in picture effects that can allow a reader to differentiate between pictures he can manipulate and those he cannot.

Picture corners are a good way to indicate a picture is “placed” on the page as a “moveable” item, that the picture is different from other pictures on the page that cannot be manipulated.

You can also use the “grunge” edge effects for your design language, to indicate objects that can or cannot be manipulated. (I prefer to use grunge to indicate something unchangeable.)

There are different ways to implement the multitouch.

### Object attribute: canPinchZoom

First, you can set “canPinchZoom” to true for a picture to allow immediate manipulation on the page. In this case, the reader can pinch, zoom, rotate, and move the object without doing anything else. However, when he releases his fingers, the object returns to its original state.

### Zooming

When you make an object zoomable, then a double-tap on the object will make it zoom. Zooming means enlarging, spinning, opacity change, movement, or a color change (TC). Zooming is set by using one or more of the zoom attributes. Currently, zoom only works with pictures.

An object both can have the “canPinchZoom” attribute set and still be zooming.

If an object is zoomable, then a double-tap will “open” the object, zooming it to whatever state is should be in. Typically, this is enlarged. (There is a system setting to allow single-tap zooms, too).

Once the object is zoomed, the reader can use two fingers to pinched, zoomed, rotate, and move the object. Unlike “canPinchZoom”, the object does not return to its previous position until the reader double-taps it again.

## Testing Multitouch in the Corona SimuLator

To test multitouch, hold down a key when clicking (or releasing a touch on a tablet). The app will remember that touch and “hold the finger down”, so you can then touch and move again.

# Photobook Wordpress Publication

## Overview

* Install a plugin and enter some information into the plugin settings.
* Categorize your WordPress posts into books and chapters.
* Create a book post which contains the basic information about your book.
* Press the "Publish Book" button on the book page(?)

## General Wordpress Setup notes

*Date/Time:* Be sure the time and date settings of your blog are correct. If you are using UTC time, for example, that's the date and time that will appear on the those pages. If you change the time settings, then update a book, it is possible the book won't appear as an update because it could have a date/time setting *before* your previous date/time settings, if you choose a new date/time that is before the UTC time! I'm in San Francisco, and UTC 00:00 is ahead of me, so changing from UTC time to San Francisco time meant my a new update of my book appeared a publication 6 hours before a previous version!

## Installation and Setup

The method for making books using Wordpress begins with a plugin, "mimetic-books-api".

* Turn on Permalinks!
* Install the plugin normally in WordPress.
* (optional) Install the "Post Types Order" plugin so you can order your pages.
* (optional) Install the "Taxonomy Order" plugin so you can order your categories.
* [option] Install the "Attachments" plugin so you can add multiple attachments of all kinds to the post.
* (optional) Modify the mb-settings.ini file if needed. In particular, the "distribution\_url" value is the URL of the WordPress website which holds published books. This can be the same as the website you are using to build the book:
  + distribution\_url = "http://publisherwebsite.com/wordpress/"
  + temp\_dir\_name = "mb-api-temp"
  + packages\_dir\_name = "mb-book-packages"
  + shelves\_dir\_name = "mb-shelves"
* In the plugin settings (Settings => Mimetic Books API):
  + Activate the "Book" controller
  + If you want security, activate the Auth controller. With Auth activated, you have to use "auth" methods to make things happen, and that requires programming. You cannot simply enter a URL and expect a result. I think….
  + Enter your publisher ID, *which is the WordPress login name in the publisher website!*
  + Choose a book theme — the templates that use the WordPress data to create the book.
  + *Ignore API base. It is left over from some original code, and I don't think it is needed.*

## Set Up Your Blog

You must set up categories for your book and its chapters, then a book page with publishing information about your book.

### Unique ID

You will need a unique ID code for every book. An ISBN number is ideal, but you can use any code that no one else will use. *You can only use lowercase letters, numbers, and hyphens.* Ideas for unique ID codes:

* + *book-yourcompanyname-yourbookname-date  
    Example: book-mimeticbooks-gonewiththewind-19360630*
  + *YourCompanyName-date  
    Example: mimeticbooks-1936-06-30*

### Categories

You use categories to select which pages go in the book, and how they are organized into books and chapters.

* Choose the "Categories" tab, under "Posts", in the WordPress dashboard.
* Create a new category for your book.
  + *Name:* Name the category with the name of your book, e.g. "Gone with the Wind".
  + *Slug:* Enter the unique ID of this book (just like a ISBN number) in the slug, prefixing it with "book\_". So, if your book code is "12345", the slug should be "book\_12345". We have to use the "book\_" prefix because WordPress might otherwise rename the slug. *The slug must be unique, and if you use a slug in use by a book or post, much confusion will ensue!*
  + Add a description, if you wish. It will not be used by the publishing system.
  + *Save:* Save the category.
* Create chapter categories for the book.
  + Create a new category and name it "GWTW-Chapter 1".
  + Set the "Parent" popup menu to the category that is your book title, e.g. "Gone with the Wind". This is how the system knows that this chapter is part of your book!
  + The Slug will be automatically set, e.g. to "gwtw-chapter-1", but you can change this. With chapters, we don't really care what the Slug is. *Just don't use an existing book or post slug!*
* If you installed the plugin, " Taxonomy Order", you can rearrange the order of categories by going to Posts->Taxonomy Order.

## Publishers

You should have at least one "publisher" in your system. A publisher information page is a page (not a post) that has the "publisher ID" field filled. If that field is empty, the page will not be considered a publisher information page.

Fill out at least the Title, Content, and Publisher ID. You should also include an icon for the publisher by setting the Featured Image on the page. You must use a PNG file — do not use a JPG file!

## Build a book

Set the category of each post to a chapter in your book. It is not necessary to add the category of the post to the book category (e.g. "Gone with the Wind") — setting it to the chapter in the book is sufficient.

For example, check the "GWTW-Chapter 1" category on each of post which is in chapter 1 of your book, "Gone with the Wind."

### Chapter Order

Be sure your chapters are in order. Without a taxonomy ordering plugin, such as "Category Order and Taxonomy Terms Order", the chapters appear alphabetically: "A chapter", "Chapter 1", "Chapter 2", etc. With the plugin, the chapters appear in the order of the categories, as you order them with the plugin.

### What if you have only one chapter?

You do not need chapter categories. If you only have one chapter, just mark all your posts with the book's category, e.g. "Gone with the Wind." Don't worry about chapters.

### What about sub-chapters?

A "sub-chapter" is a category item belonging to a chapter category item. Sub-chapters appear at the end of the chapter they belong to. So, if "sub-chapter 1" is a category item under "chapter 1", then "sub-chapter 1" appears as a new chapter after chapter 1.

## Pages from Posts

Each post tagged with the right category will appear in our book as a page. The content of the post is inserted in the page template (or "Book Page Format") that you choose for that post.

### Text

The text of the post is inserted into the text element of the first available text block on your template.

The title of the post is inserted into the title element of the first available text block of your template.

You can hide the title or text by inserting "###" before it, like this: "My Title" becomes "###My Title". This is handy if you need to name posts but don't want the title to appear on the actual book page.

### Pictures

Pictures are inserted into picture elements in the template in the order they are in your post.

### Audio and Video

The first audio and video files embedded in your post is put on your page as <audiofile> and <videofile>. In your book, it can be played by clicking the play in the book navigation bar.

If your template is smart and has a conditional button (by having a "condition" attribute), you might have a "play" button appear on your page, too.

Here are the formats the WordPress can figure out:

Audio formats: mp3, aac, m4a, wav

Video formats: mp4, mpv, m4v, 3gp

*IMPORTANT: There is no easy way to tell if a file with the extension .mp4 is audio or video. So, we assume that .mp4 files are video files. If you want mp4 audio, use Apple's AAC format, which is an open-source version of mp4. From the Corona Labs website: "For highly compressed formats, such as MP3 and AAC (aka MP4), AAC is a better option. AAC is the official successor to MP3 by the MPEG Group. MP3 technically has patent and royalty issues you may need to concern yourself with if you distribute anything yourself."*

### Custom Fields

The plugin allows you to use any custom fields in WordPress that are defined in the theme you are using. If the theme has the fields, they will appear in the "Custom Fields" box in your post.

These fields are used in the template either as text — as a label on the page, for example — or for other interactive elements, such as buttons. You could have a custom field that lists the ID's of linked pages, and buttons on the page will interpret this field to make the links to those pages.

## Send A book for publication

### Method #1: Publish from your blog

When you are ready to publish your book, go to the Mimetic Books API settings page. There, you can choose the book to publish, under "Book Publishing Settings." Click the button, "Publish Book". That's it!

### Method #2: Request publishing programmatically

The Mimetic Books API also allows other websites and programs to tell it to publish and package a book. To do so, the program must call the API. Assuming your blog is here:

http://myblog.com/wordpress/

the program must make an HTTP request like this:

http://myblog.com/wordpress/mb/book/build\_book\_package/?category\_slug=book2

The parameters tell which book to publish. You specify the book by providing the category id or category slug of your book's category. So, if your book's category is "Gone with the Wind," and the slug for that category is "gwwt", then your program would send:

http://myblog.com/wordpress/mb/book/build\_book\_package/?category\_slug=gwwt

If the ID of your category was "125", you could a send this:

http://myblog.com/wordpress/mb/book/build\_book\_package/?category\_id=125

The WordPress blog will build the book and package it into a ready-to-download tar file, with the necessary icon, poster and info files.

## Shelves

The shelves file is a JSON file of the list of available items on a website, available for download.

## Corona File structure

\*This will change, as we move toward the ability to have multiple books. \*

### iOS File Structure

Files for a book must be collected into the \_user folder.

In particular, the book.xml and settings.xml must be in the \_user folder.

Other files should be in folders inside the \_user folder, e.g. \_user/page/image/mypageimage.jpg or \_user/chapter/audio/myfile.mp3

System UI graphic elements are in the \_ui folder. Do not change these. Instead, add your own in a \_ui folder inside of the \_user folder.

### Android File Structure

I am not sure if you can use folders with Android, but I don't think so.

## Custom Files

Files to customize the app as a whole should be in the \_custom folder. These include general app settings and titlebars graphic files for the library.

The custom settings.xml file in the \_custom folder looks exactly like the general settings.xml file, but it only has to have values that you wish to change in the main settings. Values in this custom file overwrite values in the main settings.xml, and new values will be added to the settings (not sure why you’d do that).

Files that could be in \_custom:

settings.xml

shelves-titlebar.png

shelves-titlebar@2x.png

## ISBN Numbers

You don't need an ISBN, but you can get one. These are basically used as a product identifier. The ISBN is mostly useful for distributors and retailers to locate and track product.  I'm not sure how valuable they are for an IPad app, but it puts it into the *Books In Print* database.  You get them from the ISBN agency here, if you are interested : https://www.myidentifiers.com/

## Build Files

These are files that are used by Corona and iOS to build the app, but are not really inside the book.

As part of your project, keep a copy of all your build files apart from your working Corona files. If you change projects, these files will change and they cannot be stored in the Corona project apart from the .lua program files.

|  |  |
| --- | --- |
| Corona files | build.settings |
| Icons | Icon-72.png  Icon-512.png  Icon-512@2x.png  Icon-Small-50.png  Icon-Small.png  Icon-Small@2x.png  Icon.png  Icon@2x.png  iTunesArtwork |
| iOS Loading Images | Default-Landscape.png  Default-Portrait.png |

# Building the App

## Tasks to build and upload an app to the itunes App store

There are many settings that have to adapted to each use of the photobook application.

* Be sure there are low-resolution versions of all pictures on the website specified in the settings.xml file, in facebook.picturesBaseURL.
* Be sure the settings in settings.xml, in the facebook and twitter classes, are correct. Here you specify websites and text that will appear in Facebook and Twitter messages.
* Set the <uniqueid> element in the book.xml to a unique code. Preferably, this should match the book name, the ISBN, or something that will *never* be duplicated anywhere. The application name isn't a good idea, because once the application allows multiple books inside it, the code will not be unique to each book.
* Set the Facebook and Twitter values in the user settings.xml properly.
* If you need to localize, for a new language....do something.

## Checklist

iOS Developer Website Portal

* iOS Provisioning Profiles
* New App ID

|  |  |
| --- | --- |
|  |  |
| App ID | Mimetic Books  com.mimetic.mimeticbooks  *Bundle Seed ID: use Team ID* |
| Development Profile: | Mimetic Books Dev |
| Distribution Profile: | Mimetic Books Dist |
|  |  |
|  |  |
|  |  |
|  |  |

## Artwork

### Library Title Bar

Make a custom library title bar graphic. Put in the \_custom folder in the main directory. Set the value of *shelves:titleBarBackgroundImage* to the file name in the settings.

### Icon, DefaultLandscape/Portrait

Create icons, the DefaultLandscape and DefaultPortrait files required by iOS. You'll need a very big version for the Apple App store, too.

## Provisioning Profiles

In the Apple iOS development center, in the Internet:

* create a new app ID
* create a development profile
* create a distribution profile

## Settings

* Set the URL and related settings to link the app to a Wordpress website.
* Set the Facebook and Twitter settings, especially for URL's, tinyURL, etc.

## Build Settings file, Config File

You'll need to set the build.settings and config.lua files correctly to build with Corona. The config.lua stores important Facebook settings (must set up the Facebook app first).

## Choose a short App Name

Be sure the name of the app you choose is short enough so it looks good on the iPad/iPhone. If it is too long, it will have elipses in the middle, and it will suck.

## Facebook

See here: <http://docs.coronalabs.com/guide/social/setupFacebook/index.html>

It is *very* *important* that you use the correct provisioning profiles, even for testing, or the Facebook link won't work.

## Twitter

Nothing special, just enter the right information into the settings.xml file.

## Fonts

You can add fonts to use. Fonts with the "ttf" suffix work well. Simply put them into the main app folder, where the .lua files are. Be sure you reference them with the correct names, which may not be the same as the file names.

## Font Metrics

The book uses special font metrics (e.g. sizes of ascenders, descenders, etc.) to correctly position fonts. You should be sure the metrics for the fonts you use are in the system. These metrics are generated using the fontmetrics PHP script.

Place the fonts you want to use in the "fonts" folder, in the fontmetrics script folder. On a web server with ImageMagick installed, run the PHP script. Then, copy the resulting output into your fontmetrics.txt file, in the Photobook main folder, next to the fontmetrics.lua file.

In the future, you should be able to include this file with your book files.

## Mapping an Image

You can have a Google Map popup appear for the main image on a page by including a search term in the <location> page element.

If you are geotagging your pictures, this value will probably be in latitude and longitude, e.g. 39.001693, -119.956026.

You might also have the GPS coordinate in degrees, e.g. +39° 0' 6.09", -119° 57' 21.69", but using this style makes Google Maps show no only a marker for your coordinates but also the nearest known map feature — not always the effect you may want. A green arrow marker marks the position of the coordinates searched for, and a typical red pointed marker that is placed on the nearest road or geographical feature in Google's database.

You can use any search value, it turns out. Whatever you enter in the <location> element, that is what Google Maps will search for. Therefore, you can even use a city-state-country value, such as "Las Vegas, Nevada" or "Paris, France", and you will get a map of that place.

## How to share demo apps with others

To share a demo app with someone, you must add their device ID to the developer list (max. 100 users).

After adding the device at the Apple iOS developer site, and if XCode is managing the development provision profile, *you must refresh the provisions page in XCode, or the provision will not know the new device.* After refresh, the ream provisioning profile in XCode will indicate one more device.

The tricky part is creating an .ipa file, since the app is not written in Xcode. Instead, here's how:

* Build the app for a device using Corona SDK
* Open iPhone Configuration Utility
* Drop the newly built app in the Library : Applications pane, but not into a device applications pane.
* Now, select that app, and click the Share icon in the bar at the window top. An email will open, with the app in the proper ".ipa" format. Drag the file from the email to your desktop, where you can now send it however you wish.

Here is what you need to tell the client/tester:

### 1) Sending Your Device ID to a Developer

Before a developer can send you an application for testing, they must register your device with Apple under their application-testing program.

To send your device ID to a developer for test-program registration:

* Launch iTunes.
* Connect your device to your computer.
* Select the device in the Devices list.
* In the Summary pane, click the Serial Number label. It changes to Identifier.
* Choose Edit > Copy.

Email your device identifier to the developer. Be sure to include your name and device name in the email.

### 2) Installing an Application for Testing

After being registered in a developer’s testing program, the developer sends you an archive of the test application. You need to install the archive into iTunes to run the application on your device.

To install the test application on your device:

* In the Finder, double-click the application archive, <Application\_Name>.ipa.The application appears in the iTunes Applications list.
* Sync your device.If the version of iOS on your device is earlier than the test application can run on, you need to update your device with the current release of iOS.

## Exporting Chapters from Adobe InDesign

With the "Tag XML for Photobook" script, you can export a book from Adobe InDesign.

Put a copy of the file "Tag XML for Photobook.jsx" in your InDesign scripts folder, probably here:

~/Library/Preferences/Adobe InDesign/Version 7.5/en\_US/Scripts/Scripts Panel/

There is a demo document that is set up with object styles, XML, etc.

### Setting up your InDesign File

Your InDesign file should be the size of the screen you're designing for. For an iPad, that would be 1024x768 pixels (also for the iPad 3 with the retina display — don't double the size).

You'll need to make some changes to your InDesign file to make it work. Here's a list of what you need to do:

* Load Object Styles from demo document
  + It is important that the styles be properly defined. The "Split Text" style must NOT have any attributes, or all text frames will modified by its settings. So, be sure your styles don't overlap with the Photobook styles.
* In the Tags panel, load the photobook.dtd
* View:Structure:Show Structure
* Right-click "root", and choose Tag Element. Choose "book" as the root element.
* Copy the four text fields from the demo document, on the pasteboard, for title, author, uniqueid, and imagePath.
* Paste them onto the pasteboard of the first page. The tags for them should appear under the "book" tag.
* Add attribute to the "book" tag, xml:space = default. (If you validate the XML, using the lightening icon, it will suggest this kind of thing.)
* Use only RGB colors. The system will not translate CMYK colors into RGB, and you will get weird results.

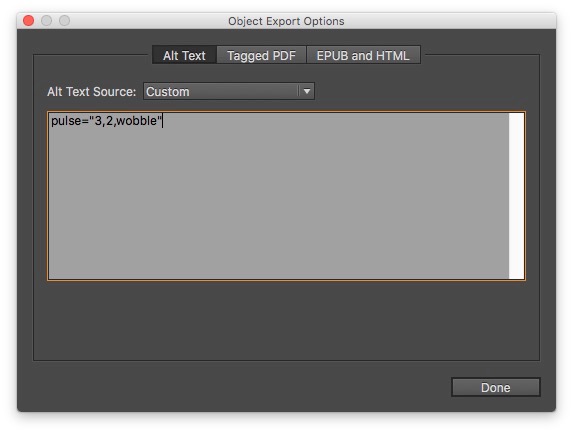
### Object Styles and Setting XML Attributes of Objects

You can use object styles to add XML attributes to elements such as images, text blocks, etc. The Javascript script that converts InDesign to XML will create some object styles you can use. For example, there is a “Zoom to Center” object style, which makes a picture zoom to the center when the reader double-taps it.

You can use an object’s “Export Options:Alt Text” to create XML attributes. Better, you can set up attributes by setting them in the “Export Options” of an object style. From the menu bar, choose “Object:Object Export Options. Choose “Alt Text”, then choose “Custom” for the “Alt Text Source”.

Enter your attributes in XML fashion, e.g.

pulse=“3,2,wobble”

You can set up styles for your pictures and text, so you don’t have to set these custom attributes by hand every time. You can add XML attributes to any object with an object style by setting the Alt Text (near the bottom of the object style editing window).

### Captions and Overlays

Captions must be on a separate "Captions" layer.

Overlays must be on a separate "Overlays" layer.

### Hyperlinks (Actions)

You can create hyperlinks in text either by using InDesign's hyperlinking or by using a special character style.

Select some text, then using the Hyperlinks panel, create a new hyperlink. Be sure it uses “Link To: URL”. You’ll see that choice if you edit the hyperlink.

The default kind of hyperlink is a “webpopup” action. This is what you get if you mark some text as a hyperlink and provide a URL, e.g. “http://mywebsite.com/”. If you do nothing else, the URL will be shown in a popup web page.

To use a hyperlink to fire an action, the URL of the link should be the action name, e.g. "toggleCaptions".

For example, to jump to the page with id = “khalid-eid”, enter

goPageByID,khalid-eid

as the URL of the hyperlink.

*Hint: It's easier to use a character style in InDesign for simple actions than to enter a hyperlink code each time for an action like “show caption”.*

**Prefix the name of the character style for such an "action" style with "action\_".** So, if you want to make some text be a hyperlink that toggles the visibility of the caption box, create a character style called "action\_toggleCaptions", and mark your text with this character style.

## Custom Fields

You can use custom field variables in your InDesign file. A custom field variable is replaced with some value from the book or the page, such as the page number or the book title, or with some custom value you set in your XML file in the <fields> element.

A field looks like this in the text: {{myfield}}

You can also use these variables as button parameters,

For example you might write,

## Buttons

The *params* attribute of a button comes from the InDesign value for the URL or the bookmark, or for a “goPage” action, the target chapter number and page number.

### Audio buttons

When using audio buttons on a page, it is handy to have a ‘pause’ button that appears when you tap the play button. Below is how to make the ‘play’ and ‘pause’ buttons. The ‘pageaudiopause’ button is automatically hidden when audio is cancelled, e.g. by the chapter’s cancel on page change command.

I think this only works for one pageaudiopause button per page…I haven’t tried two! It might even have trouble if you make two. This was designed to play the page’s audio file, not a specific file, which would explain why it isn’t smart enough to handle two audio buttons on a page.

<buttons absolute="true">

<button pageaudiopause="true" x="321" y="610" height="44" width="44" id="Button 12" params="pauseAudio" action="" imagefile="\_ui/button-audio-pause.png" over=“\_ui/button-audio-pause-over.png"></button>

<button pageaudioplay="true" x="321" y="610" height="44" width="44" id="Button 12" params="playAudio,\*/audio/Roy Hattori Audio Interview.mp3" action="" imagefile="\*/images/chapter\_5/button-speech.png" over=“\*/images/chapter\_5/button-speech.png"></button>

</buttons>

***Above is INCOMPLETE!!!***

## Page Attributes

You can set attributes for a page by creating a text field in the “pb Settings” layer. Only the first found text field will be used — if you have more than one, it’s hard to know what will be used!

The text field should contain pairs of attributes, in one line, XML style. Use double-quotes, please. For example:

id=”my-page” contents=”true” myattribute=”myvalue”

## Exporting Chapters from Adobe Lightroom

You can use Adobe Lightroom and the LR/Transporter plugin to export entire chapters of photo pages. The plugin will create a text file, which you will open, and copy into the book.xml file. You might wish to add additional pages, such as a chapter opening page.

LR/Transporter can be found here:

http://www.photographers-toolbox.com/products/lrtransporter.php

To begin, select your ordered images in Lightroom. It is much easier to put them in the right order *before* you make your text file.

Next, choose File:Plugin Extras:Export Metadata using LR/Transporter.

Name your file chapter.txt (however, the name doesn't really matter).

Choose a directory to put the summary text file, not in the actual Corona project directory.

In the dialog that appears, in the text area called "Specify text for the head of the summary" enter "<chapter>".

At the bottom, in the text area called "Specify text for the foot of the summary" enter "</chapter>".

In the middle box, "Specify the text to add for each photo," enter the following:

|  |
| --- |
| <page>  <caption>{caption}</caption>  <date>{dateCreated[%m/%d/%Y]}</date>  <imagefile>{fileName}</imagefile>  <sublocation>{location}</sublocation>  <gps>{gps}</gps>  <city>{city}</city>  <state>{stateProvince}</state>  <country>{country}</country>  <headline>{headline}</headline>  <order>{seq[1]}</order>  </page> |

In this variation for a book file, we use the imagefile name as the page ID. Of course, a more compact ID code would more elegant, but this will work. This is useful when you want to have an page\_metadata.xml file to update your page information (see far below for this subject). *Note that the ID must be an attribute and not an element…for no good reason. It just is that way for now.*

|  |
| --- |
| <page id="{fileName}">  <caption>{caption}</caption>  <date>{dateCreated[%m/%d/%Y]}</date>  <imagefile>{fileName}</imagefile>  ...  </page> |

Here is a method for also extracting the EXIF info — lens, aperture, etc.

|  |
| --- |
| <page>  <caption>{caption}</caption>  <date>{dateCreated[%m/%d/%Y]}</date>  <imagefile>{fileName}</imagefile>  <sublocation>{location}</sublocation>  <gps>{gps}</gps>  <city>{city}</city>  <state>{stateProvince}</state>  <country>{country}</country>  <headline>{headline}</headline>  <order>{seq[1]}</order>  <shutterSpeed>{shutterSpeed}</shutterSpeed>  <aperture>{aperture}</aperture>  <flash>{flash}</flash>  <exposureProgram>{exposureProgram}</exposureProgram>  <isoSpeedRating>{isoSpeedRating}</isoSpeedRating>  <focalLength>{focalLength}</focalLength>  <focalLength35mm>{focalLength35mm}</focalLength35mm>  <subjectDistance>{subjectDistance}</subjectDistance>  <dateTimeOriginal>{dateTimeOriginal}</dateTimeOriginal>  <cameraModel>{cameraModel}</cameraModel>  <cameraMake>{cameraMake}</cameraMake>  <gpsLongitudeDecimal>{gpsLongitudeDecimal}</gpsLongitudeDecimal>  <gpsLatitudeDecimal>{gpsLatitudeDecimal}</gpsLatitudeDecimal>  <exposureBias>{exposureBias}</exposureBias>  </page> |

Finally, click the "OK" button to export your chapter text file.

Of course, you could modify this export if you use fields differently. You may want to use the "{location}" tag for the mapping location, instead of using "{gps}", or you could even use "{city}, {state} {country}" in the <location> element.

## Facebook Integration: Must have picture stored on the internet

The Facebook button in the caption bar can post a photo from the book to Facebook.

There are two ways this can happen. The first is by writing a message to the reader’s wall; the second is by actually uploading a picture. The settings file entry, “postPictureMethod” in the Facebook section, controls which method is used.

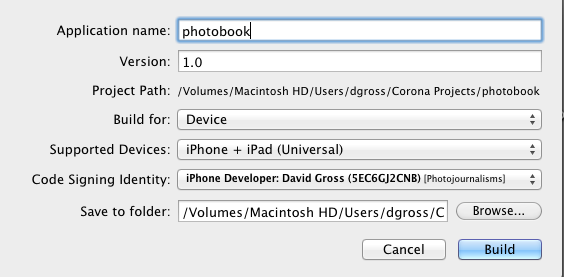
The possible values are “post” and “upload”:

<postPictureMethod value="post" />

### Important!!!

*Facebook integration is delicate! Here are some tips:*

1. **Build with a custom provisioning profile even when developing.** You *must* build the app using the bundle ID specified in the Facebook app, or Facebook won’t return the reader to the app. *This is true when developing as well as distributing!* Therefore, you must create a Development Provision Profile for your app, even if you see no other reason not to use the Team Provisioning profile. When you build in Corona SDK, the build might look like the following — note the Code Signing Identity, which is where you specify the bundle ID, is *not* the team profile (with an asterisk) but is the one I created just for this app:



1. **Be sure the picture file is specified.** If you are not getting any pictures showing up, be sure that
   1. If you are posting and not uploading, that you uploaded pictures to a website and specified the URL (see “Writing on the Wall” below)
   2. The picture name is specified either in the <imagefile> or <publicfile> tag, AND
   3. Your templates include a blank entry for <publicfile> so the page entry won’t be lost.
   4. The Facebook app ID and secret code are in your \_user/settings.xml file
   5. The Facebook app ID is in your build.settings file, in the CFBundleURLSchemes entry, beginning with an “fb”. If your Facebook App ID is “168436396551133”, then the entry will be “fb168436396551133”.

### Writing on the Wall

The "F" button allows the reader to post a message to Facebook about a picture. However, for this to work, a low-resolution version of the picture must be available on the Internet at the location specified in the settings.xml file.

Create 250px (for example) wide versions of all pictures, using identical names as in the book, and upload to a URL specified in the settings.xml, in the Facebook section.

<picturesBaseURL value="http://www.mywebsite.com/image/mybook/" />

The file names must match *either* the <imagefile> value, or be specified in the <publicimage></publicimage> entry. For example both of the following will post a photo called “myphoto.jpg” by making a link to it in Facebook from your website, specified above:

<imagefile>myphoto.jpg</imagefile>

<publicimage>myphoto.jpg</publicimage>

The result will be, in your Facebook entry, the image from:

http://www.mywebsite.com/image/mybook/ myphoto.jpg

### Uploading Pictures

To upload pictures, set postPictureMethod in your settings file to:

<postPictureMethod value="upload" />

For uploading to work, the app needs the files to upload to be in the same folder as the page images. For example, if the page image is

<imagefile>mypic.jpg</imagefile>

then the file for uploading must have the same name, followed by “-public” (or whatever you have the public file suffix set to in the settings file, in the “uploadablePicturesSuffix” tag in the “app” section).

For example, if your page has

<imagefile>mypic.jpg</imagefile>

then the file to uploaded will be “mypic-public.jpg”.

If you don’t have an <imagefile> tag because you are using <pictures> tags, you must set the <publicimage> tag to contain your filename. It should contain the “real” filename, e.g. “mypic.jpg” and *not* the “mypic-public.jpg” filename. The app will figure it out.

Why have a “-public” file at all? First, you will want to use smaller files for uploading than for showing. Also, you may want to watermark, or otherwise modify the uploadable file. I put a big “©” across my images.

## Books

Book files describe a book that the system can display. Book files are simple XML files, best read with the book.lua module. The book.lua module uses a simple XML parser to read the file, then does a fair amount of parsing to figure out details of the book. In a better world, the rest of the system would be happy to work with more "native" XML data, e.g. getting values from myelement.value, but I wasn't thinking that way initially.

If the book is read using the book.lua module, the book table (what Lua calls an array, I guess) gets some useful extra values.

### Book Table

|  |  |
| --- | --- |
| book.length | total page count of the book, including cover, contents, etc. |
| book.chapterCount | number of "body" chapters, i.e. not cover, contents, appendix. |
| book.contentsChapterNum | The number of the chapter used for the contents page. goToContents() uses these to find the contents page easily (see "body" attribute, below) |
| book.contentsPageNum | The number of the page used for the contents page. goToContents() uses these to find the contents page easily. Note: this is the page *in the chapter,* not from the beginning of the book. |
| book.contentsPageNumInBook | The number of the page used for the contents page. This is the page number in the book, from the beginning. Not sure how useful this is, since we can't find pages in the book table this way. |
| chapter.bodyChapterNumber | the number of the chapter among the body chapters, i.e. not counting the cover, contents, etc. |
| book.slideshowChapterNum | The chapter in which to start the slideshow. Set by setting the attribute of the starting <PAGE> with slideshowstart="true" |
| book.slideshowPageNum | The chapter in which to start the slideshow. Set by setting the attribute of the starting <PAGE> with slideshowstart="true" |

## Page Layering

The different items on a page are grouped by type (to allow easy purging to save memory). You can change the layering in the settings file, but by default, the different items on a page are built and layered as follows:

1. background
2. main image
3. animated/zooming pictures
4. text block
5. animation
6. links
7. buttons
8. caption
9. overlay

## Text Formatting

Until iOS gets proper text formatting, we have limited options.

You can format paragraphs of text. A paragraph is a block of text ending with a return, that is wrapped to fit some space.

### Indenting Lines

If a line begins with a period (.), the *first* period will be stripped off, but the rest of the line will not be trimmed of white space. For example,

. Text Is Indented

will be indented because of the spaces before the word, "Text". The spaces before the period at the beginning of the text *will* be removed. You can still start a line with periods. In the following case the periods will remain, except for the first one:

.......... Text here

You can also force a blank line by writing a line which contains only a period. It's better to use ### blank (see below), because, well, this feature may eventually change.

### Formatting Commands

These formatting commands work in the settings.xml file, too. You can use them to format captions, for example.

A command line begins with "###" and is followed by commands and parameters, separated by commas. Extra spaces are removed. Parameters can be blank.

Example: ### set, Baskerville-Bold, 18, 24, 255,100,255, 500, 0.5

Example: ### blank, 8

|  |  |  |
| --- | --- | --- |
| ### blank | | Insert a blank line |
| parameters | line height | Blank line height in pixels |
| ### setline | | Set font parameters |
| parameters **in order** | x | Set the x of the line, from the left of the text block |
|  | y | Set the y of the line, from the top of the text block |
|  | line number | Set the line to write to, starting with "1". That is, set the y based on the line count, i.e. the line to write to. For example, if the line height is 20, and the "line number" is set to 3, the next line will appear at y=60 from the top. |
| ### set | | Set font parameters |
| parameters **in order** | font | an font name the device recognizes, e.g. Baskerville-Bold |
|  | font size | font size in pixels |
|  | line height | line height in pixels |
|  | color: red | red component of RGB color (0-255) |
|  | color: green | green component of RGB color (0-255) |
|  | color: blue | blue component of RGB color (0-255) |
|  | width | column width in pixels or percent of the screen width, e.g. 50% on the iPad is 512 pixels (50% of 1024 pixels).  "reset" or "r" will restore the width to the original setting for this box. |
|  | opacity | opacity of the type (0.0 - 1.0) or in percent, e.g. 50% |
|  | case | Values are ALL\_CAPS or NORMAL.  Default is NORMAL. |
|  | alignment | left, center, or right:  Reset the font parameters to the settings specified by the enclosing item, or the system. Really, just return to the default for this text item. |
|  | spaceBefore | pixel space before each paragraph |
|  | spaceAfter | pixel space after each paragraph |
| ### style | |  |
| parameters | name | The name of a style in the style sheet. See "Text Style Sheets", below. |
| ### textalign | | Set the alignment of the text |
| parameters | alignment | left, center, or right |
| ### reset | | Reset the font parameters to the settings specified by the enclosing item, or the system. Really, just return to the default for this text item. |

## Text Style Sheets

The formatting commands, above, can be stored in a style sheet. There can be a system style sheet and a book-level style sheet, both named "textstyles.txt".

These text files store the style definitions in name-value pairs, e.g.

Normal = HelveticaNeue,16,20,255,20,20,,100%

Heading 1 = HelveticaNeue-Italic,16,20,20,255,20,,100%

Quote = BradleyHandITCTT-Bold,20,28,0,0,255,,100%

To use a style sheet in your text, use a "### style" command, from the table above.

## Special Characters in XML

You can use show standard characters in your XML by using these codes:

|  |  |
| --- | --- |
| Character | Code |
| " | &quot; |
| > | &gt; |
| < | &lt; |
| & | &amp; |

## Conditions

Some page objects can have conditions which determine how, or whether, they will appear.

For example, a button might have a condition that says that if there is a videofile for the page, then it will appear. If not, the button will not appear.

You can write a condition to test for any "top-level" object on the page. You can't test for sub-objects, like buttons inside the "button" tag, or text inside a "textblocks" tag.

Here are some of the objects you can test for (I may have forgotten something) : *videofile, audiofile, title, altTitle, textblocks, buttons, pictures, panoramas, id, creator, headline, caption, location, date, city, state, country, publicfile, backgroundfile, imagefile, overlay, etc.*

You could even add your own tag in the page XML, such as "myflag", and while it won't render on the page you can use it as a flag.

### The Condition

A condition is a string of values, separated by commas, e.g.

condition = "audiofile,is not, blank"

The first value is the name of the XML tag on the page, e.g. "audiofile".

The second value is the condition. Either "=" or "is" are the same, and they mean *equals*. Anything else is interpreted as *not equals.*

The third value is the value to compare with. Acceptable values are: blank, empty, nil, null, true, false, or a string or number.

### Multiple Conditions

You can have more than one condition in your string. The conditions will be tested and joined by "AND". That means they must all be true for the condition to be true.

### Examples

<button condition="audiofile,is not, blank" ... />

<button condition="title,is,My Title" ... />

<button condition="number,=, 5" ... />

<button condition="caption,=,empty,videofile,is not,empty" ... />

### Condition Results

Conditions generally determine whether the object with the condition will appear on the page. So, if your condition returns TRUE, then the object will appear.

Conditions test either page elements, e.g. "audiofile", or they can test attributes of the object itself.

First, we check whether the object has the value in question. If an object had an attribute "title", we would check that. If there is no such attribute, then we check the page for "title".

So,

<button condition="title,is,My Title" ... />

will check the page's title because buttons don't have title attributes. Of course, you could add a "title" attribute to the button, and then the condition would use the button's value. That would work fine because the "title" attribute would be ignored by everything else!

## How to Show Multiple Languages or alternative Texts

You can use conditions to make your book show different languages (or different texts) depending on a book value, e.g. “language”. This value can be any XML tag not in use. I suggest using “language”.

Start by setting the book’s language. For example:

<book>

...

<title>My Book Title</title>

<language>English</language>

...

</book>

Objects like buttons or pictures can have conditions, so you set those as above, e.g.

<button condition="language,is,French" ... />

For text, you add extra text into the <textblock> elements, as <text> and <title> elements, e.g.

<title><![CDATA[The Lost Boys]]></title>

<title condition="language,=,French">

<![CDATA[Les Garçons Perdus]]>

</title>

<title condition="language,=,Arabic" x=”500” width="325" height="59">

<![CDATA[<p style="text-align:right;" >فقدت بنين</p>]]>

</title>

Notice that not only can you add new text to replace the default text (which has no conditions), but you can also add other parameters, such as “width”, and they will overwriter the default parameters. In our example above, notice that the x, width, and height of the Arabic text are set. These values will override the defaults, or the template values.

This method can be used to show alternative languages in a book, depending how the book’s “language” value is set, and that could be changed with a button.

You could also use this technique to show different texts depending on other values, too. Perhaps the user bought something in the app — you could show different text based on that kind of value, too.

Or, what if the date changed, perhaps later than a fixed date (like a new year). The text could change based on that, too.

Here is a sample page with default (English), French, and Arabic text:

<page id="the-lost-boys" date="2007-07-29 14:20:48" modified="2013-05-04 00:01:02">

<title>My Page Title</title>

<altTitle>The Gross Report</altTitle>

<textblocks>

<textblock>

<title><![CDATA[The Lost Boys]]></title>

<title condition="language,=,French">

<![CDATA[Les Garçons Perdus]]>

</title>

<title condition="language,=,Arabic" x="500" width="200" y="60" height="59" absolute="true" >

<![CDATA[<p style="text-align:right;" >فقدت بنين</p>]]>

</title>

</textblock>

<textblock>

<text>

<![CDATA[<p><em></em></p>

<p><em>Shadows of War,</em> page 184:</p>

<p><em id="\_\_mceDel"> &#8230;The children take great exception to the common statements that children born and bred in war are a &#8220;lost generation&#8221;&#8230;.</em></p>

<blockquote><p>Illusion, <em>the children respond.</em> We know how we came to live this way. We can see who has and who doesn&#8217;t, who gives and who takes. We know we take better care of each other here [on the streets] than any of us would find in the places we ran from&#8230;.</p>

</blockquote>]]></text>

<text condition="language,=,French">

<![CDATA[<p><em></em></p>

<p><em>Ombres de la guerre,</em> page 184:</p>

<p><em id="\_\_mceDel"> &#8230;... Les enfants prennent une grande exception aux déclarations communes que les enfants nés et élevés dans la guerre sont une «génération perdue» ....</em></p>

<blockquote><p>Illusion, <em>les enfants réagissent. </ em> Nous savons comment nous sommes arrivés à vivre de cette façon. Nous pouvons voir qui a et qui n'a pas, qui donne et qui prend. Nous savons que nous prenons mieux soin de l'autre ici [dans les rues] que chacun de nous pourrait trouver dans les endroits où nous avons couru de ....</p>

</blockquote>]]>

</text>

<text condition="language,=,Arabic"><![CDATA[

<p style="text-align:right;">السلام عليكم</p>

<p style="text-align:right;">الأسماء حسب ظهور الصور بالترتيب</p>

<p style="text-align:right;">غدير غزال 17 ادلب</p>

<p style="text-align:right;">قمر حموي 15 ادلب</p>

<p style="text-align:right;">الاء الجادري 15 حلب</p>

<p style="text-align:right;">معتز العلي 18 ادلب</p>

<p style="text-align:right;">هبة خربطلي 18ادلب</p>

<p style="text-align:right;">غيداء غنوم 16 ادلب</p>]]></text>

</textblock>

</textblocks>

<template>AP</template>

</page>

## The Inventory

The inventory is a list of items in the book XML file which can be changed. Each has a changeable status, such as locked/unlocked, hidden/visible, etc. In particular, this is useful for recording "in-app purchases", such as unlocked chapters, pages, videos, etc.

The system tries match each item with an id in the book with an item in the inventory. Each inventory item should have an id element which should match an element in the book definition. For example, a chapter with id="myLockedChapter" can have an entry in the inventory with the same id. Items in the book do not need to have corresponding items in the inventory, and vice-versa — the book will work fine with "extra" items floating around. Not a problem.

Setting the status of an item in the book, such as a page, is done in the inventory. So, to lock page "myLockedPage" you create an item in the inventory with the id, "myLockedPage", and set a locked element, like this:

<inventory>

<item>

<id>myLockedPage</id>

<locked value="true" />

...

</item>

...

</inventory>

IMPORTANT: You must set true/false values as attributes! The following *will not work* *—* it will set the value of "locked" to the text value, "true", rather than the boolean TRUE.

<locked>true</locked>

## Locked page example

The following example shows how you can lock a page, making it unreadable, but allow it to be viewed by using a password.

First, in the chapter element, create an “inventory” element:

<inventory>

<item>

<id>lockedpage</id>

</item>

</inventory>

Next, here is your page. Note the “id” links this page to the inventory item. The page provides the password for unlocking, and it controls the initial state (“locked”) of the page.

<page id="lockedpage" locked="true" password="mypassword">

<textblocks absolute="true">

<textblock>

<text>This page is locked.</text>

</textblock>

</textblocks>

</page>

## In-App Purchases

To make an item available for purchase using in-app purchases, such as through the iTunes store, you need to create an inventory item for it.

First, identify the item in the book with an ID. Examples:

<chapter id=" lockedchapter"> ... </chapter>

<page id="lockedpage"> ... </page>

<button id="playvideo" ... />

Next, make an inventory entry for the item. Some of the item's elements are copied from the iTunes store: title, description, productIdentifier, name, and price (see note, below).

The important element is the "action" element. If it is set to "purchase", as in the example, then the item is available for purchasing.

Example:

<inventory>

<item>

<id>lockedchapter</id>

<locked value="true" />

<action>purchase</action>

<success>unlock</success>

<title>Premium Chapter</title>

<description>Tap to unlock this chapter.</description>

<productIdentifier>

com.company.app.ConsumableTier1

</productIdentifier>

<name>Premium Chapter</name>

<price>1.99</price>

</item>

...

*Important: While you can enter a price, it is a bad idea if your item will be purchased through Apple, and possibly other in-app purchasing solutions.*

*The reason is that you should use depend on the price set in the iTunes store, and not depend on a price set in the app itself. Some stores, like iTunes, used "tiered" pricing, meaning they can change the price on you. In that case, your app will show the wrong price.*

*Also, the iTunes store will send the app the price sometime after the app starts up, but not right away. If there is no Internet connection, the price will not be sent. In either case, it is possible that the app will not show the price at all because it did not get it before loading the page.*

*Conclusion: Don't use the <price>xxx</price> element. The user will learn the price when they try to buy the object anyway.*

### Inventory Item Elements:

|  |  |
| --- | --- |
| id | The unique identifier for the item. It should match the ID of an element in the book. |
| locked | true/false : if true, the item starts as locked. If false, the item is not locked, useful perhaps for games and such. |
| action  NOTE: Attributes of by the item affected, e.g. a page or chapter, control what happens. These values are: *message, id, action, locked, password* | *purchase* : This item can be purchased. If the user taps this item, he will be offered the chance to buy it.  *password* :  *message* :  *togglelock | unlock | lock* : This action will toggle the lock, or unlock, or lock the item.  *test* : |
| name | The name of the item. This should match the "Display Name" in the iTunes Connect In-App Purchases form. This value will be shown as the headline of description in the app, if it is set.  That is, below a lock button, there can be a description of the item. That description uses the "name" and "description" elements. |
| description | The description of the item. This should match the "Display Description" or "Description" in the iTunes Connect In-App Purchases form. This value will be shown as the body of description in the app, if it is set.  That is, below a lock button, there can be a description of the item. That description uses the "name" and "description" elements. |
| showdescription | Show the name and description below the locking icon.  In the settings file, if the "slideview" setting, "inventoryShowDescriptionsByDefault" is set to "true" then the description will appear by default. If it is set to false, then the description will only show when "showdescription" is true in the inventory item definition. |

## Introductions

Introductions are short videos (and maybe more, later) than can play when the book is is opened on the first page, or anytime it is reopened.

Be sure the book background color in the settings matches your video, or the transition will be a bit jarring.

<introduction playIntro="">

<video>

<filename value="\_user/video/intro.m4v" />

<width value="1024" />

<height value="768" />

<x value="center" />

<y value="center" />

<removeWhenFinished value="true" />

<seek value="0" />

<stroke value="0" />

</video>

</introduction>

## Book XML Coding

A book is described by an XML file. The book is divided in chapters, which contain pages.

IMPORTANT: You must set true/false values as attributes! The following *will not work for a true/false setting* *—* it will set the value of "locked" to the text value, "true" rather than to the boolean TRUE.

<someTag>true</someTag>

Instead, to set the value of a tag to a boolean true/false, such as when locking an item, set it in the attributes, like this:

<someTag value="true" />

Some of the book code is written as attributes, not as child elements. Attributes of an element are the values written inside the first tag, e.g. <chapter hasCaptions="false">

### Comments:

|  |  |
| --- | --- |
| <!-- enter comment here -->  <ignore></ignore>  <comment></comment> | Anything in unknown tags is ignored. Useful to store comments, or stuff you aren't sure you want to delete, when testing or designing. |

### Book Attributes in the XML:

|  |  |  |  |
| --- | --- | --- | --- |
| title | | Title of the book. | |
| author | | Author of the book. | |
| uniqueid | | Unique ID used to save book use values, such as current page for reopening, or status of locked chapters, etc.  ***IMPORTANT: A book in the \_user folder, that is built-in to the app itself, must have the uniqueid = “default”*** | |
| introduction | | This is a table element, which can contain a video to play when the book is opened.  Attributes: | |
| attributes | playIntro | | "open" : play the intro when the book is opened.  "start" : play the intro when the book is opened to page 1 of chapter 1, i.e. probably only the first time.  *Any other value, or no value, will not play the intro.* |
| elements of introduction | |  | |
| <video>...</video> | | A video to show when showing the introduction. | |
| attributes | filename | | Full path to the video to play, e.g. "\_user/video/myvideo.m4v" |
|  | width | | width to display the video, either pixels or % (of the screen width) |
|  | height | | height to display the video, either pixels or % (of the screen height) |
|  | x | | x position from top-left of the top-left corner of the video, or  left/right/center |
|  | y | | y position from top-left of the top-left corner of the video, or  top/bottom/center |
|  | removeWhenFinished | | "true" or "false". If true, the video is removed from the screen when it is finished.  *Default is true.* |
|  | seek | | Millisecond (?) of the video to start at. |
|  | stroke | | Thickness of stroke drawn around the video, default is no stroke. |

|  |  |
| --- | --- |
| hasCaptions | "true" or "false" (default is true)  If true, then if a page has a caption, it can be seen. If not, a message will be shown, "No Caption", if the reader tries to swipe to see the caption.  If false, no caption will be shown even if one is on the page, and no message will be shown to indicate no caption |
| hasOverlays | "true" or "false" (default is true)  If true, then if a page has a overlay, it can be seen. If not, a message will be shown, "No overlay", if the reader tries to swipe to see the overlay.  If false, no overlay will be shown even if one is on the page, and no message will be shown to indicate no overlay |
| canUseNavigation | "true" or "false" (default is true)  If true, a tap will show the navigation bar. If false, the navigation bar will hide if it was showing, and tap will not show it. |
| body | "true" or "false" (default is true)  If true then the chapter will be counted as main chapter (i.e. not a forward or contents or appendix), to arrive at the total count of main chapters for the book. |
| notInContents | The chapter will not be listed in the automatically built table of contents. |
| tocSpaceAfter | Space after the automatic entry in the table of contents, measured in pixels or percent of the height of the screen.  Examples:  tocSpaceAfter="3%"  tocSpaceAfter="20" |
| showunlockbutton | If the chapter is locked, show an unlocking button. The default is "true", so this useful to hide the locking button by setting it to "false". |
| pickerLabelGroupBy | Override settings file value for pickerLabelGroupBy, which determines how labels are created in the navbar.  *year/month/day* : if the value is one of these, then labels will be created from the date element of the page, and pages will be grouped by either year, month, or day.  For example, if the value is "year", and there are two sequential pages with the dates "2001-05-11" and "2001-12-21", there will be a marker in the navbar for 2001 for the first page. The second page will have no marker in the nav bar because it is from the same year.  *text :* if the value is "text", then the page label element is written in the nav bar. Sequential repeating labels are shown only for the first in the series.  *Possible values : year, month, day, text* |
| label | The label of the page as shown in the navigation bar, when the settings for labels is set to show them.  Labels can be grouped, depending on settings in the navbar.settings.  **The pickerLabelGroupBy setting can make your labels disappear. If you are grouping by “year” then text labels won’t appear…very confusion!**  Override the settings value in the chapter itself. |
| mapPageID | The page ID of the map page, linked to the icon in the navigation bar. This overrides the main map page settings, so you can have a map in a chapter. |
| contentsPageID | The page ID of the chapter's table of contents, if it has one. This will be overwritten by a page having the attribute "chaptercontents". |

### Chapter Attributes in the XML:

|  |  |
| --- | --- |
| ignoreExtraSourceElements | ***This applies only to templates!***  If this attribute is "true" then any elements on the page source that are *not* in the template will be ignored.  So, if you have some text in your book.xml source file, but the template does not have any textblocks defined, the text will not show up.  If this attribute is missing or "false", then such elements from the source book.xml file will appear, but they will be formatted according to settings. |
| contents | "true" or "false" (default is false) The page with this attribute is recorded as the contents page for the book.  goToContents() will jump to this page. |
| chaptercontents | "true" or "false" (default is false) The page with this attribute is recorded as the contents page for the *chapter* (not the book!).  Setting this value will override the chapter setting for the same functionality, "contentsPageID". In other words, you can set this functionality in either place.  goChapterContents () will jump to this page. |
| hasCaptions | "true" or "false" (default is true) |
| hasOverlays | "true" or "false" (default is true) |
| canUseNavigation | "true" or "false" (default is true) |
| contentstitle | If this is set, then this page will appear in the table of contents, with the value of this attribute. Example: contentstitle="My Page Title" |
| textblockalpha | A value between 0 and 1, where 0 is transparent and 1 is opaque |
| backgroundcolor | RGB or RGBA color of the page background, which would override the default page background color. The string is comma-separated, and the alpha is optional. The alpha value is 0-100.  Examples:  Red at 50% alpha:  Backgroundcolor="255,0,0,50"  Dark grey:  Backgroundcolor="30,30,30" |
| number | The number of the page in the entire book, starting with 1. The cover is generally page 1. |
| slideshowstart | "true" or don't use it.  This page will be the first slide of the automatic slideshow, if the reader starts the show at a page *before* this page. |
| skipinslideshow | “true” or don’t use it.  This page will not appear in the slideshow. It will be skipped over, going to the next page. |
| showunlockbutton | If the page is locked, show an unlocking button. The default is "true", so this useful to hide the locking button by setting it to "false". |
| tocSpaceAfter | Space after the automatic entry in the table of contents, measured in pixels or percent of the height of the screen.  Examples:  tocSpaceAfter="3%"  tocSpaceAfter="20" |

### Page Attributes in the XML (See above for most of these):

### Book Elements:

|  |  |
| --- | --- |
| <comments> | anything inside the <comments>...</comments> is ignored. |
| <author> | The author's name. Can be used in various messages, etc., generated by the software |
| <uniqueID> | A unique ID to identify the book. Must be UNIX writeable because it is also used to mark configuration/data files as pertaining to the book |
| <chapter> | contains pages. Chapters are loaded/unloaded into memory, and may have a visual cue that the reader has changed chapters, e.g. a fade in and out. |

### Chapter Elements:

|  |  |
| --- | --- |
| <title> | the title of the chapter, as shown to a reader. |
| <id> | a unique ID in the book to identify the chapter. If a chapter has no title, it can still have an ID to help identify it or refer to it in some data structure. Then, if the order of chapters change, the chapter remains identified. |
| <index> | the number or index of the chapter to be shown to a reader, e.g. 1 or A or XXI. While a chapter has an internal number, indicating its order in the data structure, the index is the reader's name for it. If it is used in boilerplate text, one might have "One", for example, for chapter 1. |
| <page> | a single screen, or page, in the book. |
| <audiofile> | An audio file that will be played during a slideshow. This can change with each chapter. |

### Page Elements:

(If the page contains a "textblock" element, it is may have an opaque background (in the settings file), and cover up the image. That doesn't mean the image isn't there, but it is hidden)

|  |  |
| --- | --- |
| <title> | the page title |
| <id> | a unique ID in the book to identify the page. (Same as the attribute) |
| <creator> | the name of the creator, i.e. the credit for the caption |
| <headline> | The headline, or title, of the caption.  The proper IPTC field is "headline," for the title of a picture, not "title", which means: "A shorthand reference for the news object. While a technical identifer goes to an identifier element, Title holds a short verbal and human readable name. Title is not the same as Headline." |
| <caption> | a caption which can be shown/hidden in an automatically constructed pane. The caption is automatically formatted with other page info, such as date, title, location. |
| <location> | A string to search on for a mapping website. This could be GPS coordinates of the image, or city/state/country, or anything that Google Maps can find. |
| <date> | date of the main image for the page, used in the caption. |
| <city> | for the caption, the city of the location of the image. |
| <state> | for the caption, the state of the location of the image. |
| <country> | for the caption, the country of the location of the image. |
| <publicfile> | An JPG image filename, e.g. \_user/page/myfile.jpg, which can be uploaded to a public place, such as Facebook.  Such files should be named the same as the main image file for the page, but with  "-public" at the end, e.g. "myfile-public.jpg"  You can change the suffix from "-public" to something else in your settings files. |

|  |  |
| --- | --- |
| <backgroundfile> | an image filename, e.g. myfile.jpg, which is shown behind everything else on the page, in the center. Usually, this is screen-sized. |

|  |  |  |
| --- | --- | --- |
| <imagefile> | | an image filename, e.g. myfile.jpg, which is display in the center of the page |
| attributes | x | distance from the left of the screen to the top-left of the text, e.g. 100, or "left", "right" or "center" |
|  | y | distance from the top of the screen to the top-left of the image, e.g. 100, or "top", "center", or "bottom" |
|  | height | height in pixels or percent of the image |
|  | width | width in pixels or percent of the image |

|  |  |  |
| --- | --- | --- |
| <audiofile> | | an audio filename, which can be played with a button in the caption pane. |
|  | autoplay | true/false: If true, the sound will start playing when the page is opened. |
|  | replaceCurrentAudio | true/false: If true, then this audio will stop whatever was playing. If not, the current audio will continue. |
|  | loops | Number of times to loop the audio.  0: play once  n: play once and loop *n* times  -1: loop indefinitely |

|  |  |
| --- | --- |
| <videofile> | an video filename, which can be played with a button in the caption pane. |

|  |  |  |
| --- | --- | --- |
| <overlay>  (SIMPLE VERSION) | | an image filename. The overlay appears over the main image, used to add more information. |
| attributes | overlayAlpha | the alpha (transparency) of the overlay, to override the main config setting. Values are from 0% to 100% |
|  | overlayImageAlpha | The alpha of the image used for the overlay. Not the alpha of the "paper" background of the overlay, controlled by "overlayAlpha", above. |
|  | overlayY | distance from the top of the screen of the top of the overlay, it's "y" value.  To vertically center the overlay, use "center" instead of a number. |
|  | backgroundImageFile | The file name of the background image file to use behind the overlay. Set this only if you want to override the system settings.  You must use the full page, e.g. \_user/myoverlaybackground.png  *Default is to use the settings file value.* |
| <overlaytext> | | a block of text that appears in the overlay |
| attributes | x | distance from the left of the screen to the top-left of the text, e.g. 100  To horizontally center the textblock, use "center" instead of a number. |
| <overlaytitle> | | Text that appears as the title of an overlay. |
| attributes | x | distance from the left of the screen to the top-left of the text, e.g. 100  To horizontally center the textblock, use "center" instead of a number. |
| <overlay></overlay>  (COMPLEX) | | Tag to open an overlay element in the XML. |
| attributes | backgroundcolor | Color of a background rectangle in the as a comma-separated RGB, e.g. "10,20,10" or an RGB-A value, e.g. "10,20,10,50%" |

|  |  |  |
| --- | --- | --- |
| grid | | Grids can appear on the page, or inside an overlay. |
| attributes | x | The horizontal position of an object grid in the overlay. If an integer, then measured from the left side. Other values are "left", "right", "center".  The numeric position is offset by the overlayLeftPadding or overlayRightPadding in the settings. |
|  | y | The vertical position of an object grid in the overlay. If an integer, then measured from the top side. Other values are "top", "bottom", "center".  The numeric position is offset by the overlayTopPadding or overlayBottomPadding in the settings. |
|  | width | Width of the grid, either the pixel width or a percentage of the enclosing element, e.g. an overlay. |
|  | height | Width of the grid, either the pixel height or a percentage of the enclosing element, e.g. an overlay. |
|  | direction | The direction the grid is built, either horizontally or vertically. Values are "rows" or "columns". |
|  | maxcount | Number elements across or down, depending on the direction, set by "griddirection". |
|  | padding | Padding between elements in the grid. |
|  | itemwidth | Width of a grid element. An rectangular object will be forced into this width. |
|  | itemheight | Height of a grid element. An rectangular object will be forced into this height. |
|  | itembackgroundpadding | Padding in pixels around the object when enclosed by a rounded rectangle, like a slide frame. |
|  | itembackgroundcolor | Color of the frame around the grid object, in RGB or RGB-A, where is a percentage or 0-255 |
|  | itembackgroundcorner | The radius in pixels of the corner of the item frame. If 0, then the frame will have square corners. |
|  | backgroundcolor | Color of the background of the entire grid object, in RGB or RGB-A, where is a percentage or 0-255 |

|  |  |
| --- | --- |
| <textblocks> | Contains the textblocks for the page. |

|  |  |  |
| --- | --- | --- |
| <textblock>  or  <text> when inside <textblock> | | a block of text that is wrapped to the screen. Returns in the text indicate line breaks. |
| attributes | x | distance from the left of the screen to the top-left of the text, e.g. 100 or 10%  To horizontally center the textblock, use "center" instead of a number. |
|  | y | distance from the top of the screen to the top-left of the text, e.g. 100 or 10%  To automatically position the text, use "top", "center" or "bottom" |
|  | class or style | Set the default style for the text block.  Apply the named paragraph style to the block of text. This will apply to unstyled text, and subsequent styling using ### tags or <p class=”…”> tags will override this setting.  Example:  <text class="Title"/> My Title </text> |
|  | width | Width of the textblock in pixels or percent of the screen width |
|  | height | Height of the textblock in pixels or percent of the screen width |
|  | font | the font name used by the system, e.g. " Futura-CondensedExtraBold" |
|  | fontsize | size in pixels of the font face, not points |
|  | lineheight | height of a line in pixels, or as a percent of the fontsize, e.g. 100% is the same as the fontsize. |
|  | textcolor | Text color, as a comma-separated RGB, e.g. "10,20,10" or an RGB-A value, e.g. "10,20,10,50%" |
|  | alpha | alpha of the text in percent, e.g 30% |
|  | shadowalpha | alpha of the text shadow in percent, e.g 30% |
|  | textalign (or alignment) | Alignment of the text inside the textblock area: "left", "center", "right".  *Oddly, this alignment is does not move right-aligned text to the right side of the text block area! For that, use align, below. Weird, huh?* |
|  | top | Top margin inside the area allocated for the text. Either pixel amount, or a percentage of the screen, e.g. "5%" |
|  | left | Left margin inside the area allocated for the text. Either pixel amount, or a percentage of the screen, e.g. "5%" |
|  | bottom | Bottom margin inside the area allocated for the text. Either pixel amount, or a percentage of the screen, e.g. "5%" |
|  | right | Right margin inside the area allocated for the text. Either pixel amount, or a percentage of the screen, e.g. "5%" |
|  | backgroundcolor | The textblock background color, as a comma-separated RGB, e.g. "10,20,10" or an RGB-A value, e.g. "10,20,10,50%" |
|  | stroke | The thickness of the stroke around the textblock in pixels. |
|  | strokecolor | The textblock strokecolor color, as a comma-separated RGB, e.g. "10,20,10" or an RGB-A value, e.g. "10,20,10,50%" |
|  | isContents | If “true” then the text field inside the <textblock> has the attributes for the table of contents.  *Example:*  <textblock isContents=”true”>  <text>CONTENTS HERE</text>  </textblock>  or  <textblock>  <text isContents=”true” x=”100” y=”150” absolute=”true”>CONTENTS HERE</text>  </textblock> |
|  | frame | (integer) If this has a value, then add a frame around the image with the specified thickness. |
|  | frameStyle | Set the frame style, if frame is set (above).  Values:  “solid” : A solid stroke centered on the image edge  “thin-thick” : 25% inner stroke, 50% out, with 25% padding  “thick-thin”: 50% inner stroke, 25% out, with 25% padding |
|  | frameColor | The frame color, an RGBa color string, e.g. “40,40,50,50%” or “10,30,50,128” |
|  | yAdjustment | (number) The yAdjustment lets you shift the textblock up and down in the frame. This is useful if the font metrics for your font are wrong, or you using all caps. When you try to vertically center, your text will look wrong, but the yAdjustment let’s you tweak the vertical centering. |

|  |  |  |
| --- | --- | --- |
| <texttitle>  or  <title> when inside <textblock> | | the page title, show above the text block |
| attributes | x or align | "left", "right", "center", an integer, or a percentage.  If x is an integer, the title is position x pixels from the left of the text block box.  If x is a percentage, the title is position x% *of the screen width* from the left of the *text block box.*  Alignment of the title text in the area of the textblock: values are left, right, center.  This is different from how two lines of the title would align together, e.g. on the right, which is set by "textalign", below. |
|  | y | distance from the top of the screen to the top-left of the text, e.g. 100  To vertically center the textblock, use "center" instead of a number. |
|  | font | the font name used by the system, e.g. " Futura-CondensedExtraBold" |
|  | fontsize | size in pixels of the font face, not points |
|  | lineheight | height of a line in pixels, or as a percent of the fontsize, e.g. 100% is the same as the fontsize. |
|  | alpha | alpha of the title text in percent, e.g 30% |
|  | textcolor | Text color, as a comma-separated RGB, e.g. "10,20,10" or an RGB-A value, e.g. "10,20,10,50%" |
|  | align |  |
|  | textalign (or alignment) | (Use ### textalign, …) instead to position text inside the text title.  ~~Alignment of the text in the title: "left", "center", "right"~~  ~~For example, if the title was two lines, you could align them to the right of the space they take up, even if the entire title text is, as a group, aligned to the left in the text block area!~~  ~~Example of textalign="right" in the title.~~  ~~My Title Over Two Lines~~  ~~The body text is here. The body text is here. The body text is here. The body text is here. The body text is here. The body text is here. The body text is here. The body text is here. The body text is here. The body text is here. The body text is here. The body text is here.~~ |
|  | fixedheight | Fixes the height of the title to settings height, rather than enlarging it to accommodate extra text. Normally, you would want it to grow if there is a lot of text, but for some graphic design purposes you might wish to keep it smaller. |

|  |  |  |
| --- | --- | --- |
| <pictures> | | a list of pictures, which can be zoomed in and out |
| attributes | x | Distance in pixels of the top-left corner of the picture block from the **center** of the screen. If x=0, the picture block top-left corner is in the middle of the screen. Or, x can be a relative position on the screen, values are : left, right, center.  “left” makes the image flush-left.  “center” centers the image on the page.  “right” makes the image flush-right.  Typically, you will use "left" to position the block on the left side. Use the x values of the <picture> elements to position them from the left. |
|  | y | Distance in pixels of the top-left corner of the picture block from the **center** of the screen. If y=0, the picture block top-left corner is in the middle of the screen. Or, y can be a relative position on the screen, values are : top, bottom, center.  Typically, you will use "top" to position the block at the top. Use the y values of the <picture> elements to position them from the top. |
|  | absolute | If absolute = "false" then the pictures block uses the textblock margins from the settings file: textblockBkgdTopMargin, etc.  If absolute="true" then the pictures block measures from 0,0 of the screen.  Default is "true"!  To get the pictures block at the top/left of the screen, set absolute="true", x="left", y="top". |
|  | top, right, left, bottom | You can set the margins for the pictures block by setting top, right, left, and bottom.  Example: top="100" |

|  |  |  |
| --- | --- | --- |
| <picture> | | a picture on the page |
| attributes | filename | an imagefile for the picture, e.g. "mypicture.jpg" |
|  | x | pixel distance from left inside the block of pictures, e.g. "100", or "left", "right" or "center" placement in the screen. |
|  | y | pixel distance from top inside the block of pictures, e.g. "100", or "top", "bottom", or "center" placement in the screen. |
|  | ~~center~~ | ~~"true" means the initial, or unzoomed, picture will be centered in the screen.~~ |
|  | time | time in milliseconds for the entire animation. The animation eases out slowly near the end. E.g. 1000 is one second. |
|  | width | width of the picture as it appears on screen |
|  | height | height of the picture as it appears on screen |
|  | sizeToFit | If "true" then the image is resized to fit proportionally inside of the dimensions, width x height. If either height or width is missing, the image will be proporationally scaled to fit the known dimension.  *For example, if width="100" then the image will be scaled to 100 pixels wide. If the height="50" the image will fit inside of 100x50 pixels.* |
|  |  |  |
|  | scale | scaling of the picture as it appears on the screen, where 1 is actual size. Usually this is *not* used if the width and height are set, above.  You can also use “fit” and “fill”. Fit scales to fit in the screen (maximizing the image without cropping), and fill scales to fill the screen (potentially cropping the image). |
|  | alpha | alpha (transparency) of the picture as it appears on screen, 0.0 - 1.0 |
|  | rotation | rotation of the picture, where 0 is not rotated, and 0 is the top of the circle. Example: "-45" would rotate the picture 45° to the left. |
|  | ~~referencePoint~~ | The reference point ("TopLeft", "TopCenter", "TopRight", "CenterLeft", "Center", "CenterRight", "BottomLeft", BottomCenter", "BottomRight".  ~~The picture is always anchored using the top-left. That is, the x,y refer to the top-left corner of the picture.~~ |
|  | showOnFrontstage | “true” or “false. Default is “true”.  If true, then when a picture is zoomed (by tapping, for example), it will be moved to “front stage”, a layer on top of everything else. Usually, you’ll want to dim everything below with “dimBackground”, below. |
|  | zoomedX | pixel distance from the top-left of the screen to the *center of the image* when the image is zoomed. Not such a weird decision when you actually use it. |
|  | zoomedY | pixel distance from the top-left of the screen to the *center of the image* when the image is zoomed. Not such a weird decision when you actually use it. |
|  | zoomedScale | scaling of the picture as it appears when zoomed.  “fit” zooms to fit the image inside the screen without cropping.  “fill” zooms to fill the screen, potentially cropping the image.  Number or percent: Either 1 or 0.5 or 100% or 50% work. The number is the scale or percentage of the original file dimensions.  To make an image "pop up", you could have the scale be .25, and then have the zoomedScale be 1.0. |
|  | zoomedAlpha | alpha (transparency) of the zoomed picture, 0.0 - 1.0 |
|  | zoomedRotation | rotation of the picture when zoomed, where 0 is not rotated, and 0 is the top of the circle. Example: "-45" would rotate the picture 45° to the left. |
|  | zoomedColor | The color of the object when zoomed. (NOT YET IMPLEMENTED) |
|  | ~~zoomedReferencePoint~~ | ~~The side of the object to align to, for the final zoomed position. Default is "Center". Values are:~~  ~~TopLeft, TopCenter, TopRight~~  ~~Center or CenterCenter~~  ~~BottomLeft, BottomCenter, BottomRight~~  ~~Note: The capitalization is important!~~  ~~Example: a box of size 50,50 with "center" alignment zooms to 100, 100. The left edge of the box will appear at 75 on the screen, because the center of the box is at 100.~~ |
|  | noShadow | If noShadow="true" then there will no drop shadow behind the picture element. The default is “true,” for no shadow. |
|  | zoomedYAlignment | The side of the object to align to, for the final zoomed position. Default is "center". Other values are "top" and "bottom". If an object is aligned "center", then the coordinates of the object are measured from the center.  Example: a box of size 50,50 with "center" alignment zooms to 100, 100. The top edge of the box will appear at 75 on the screen, because the center of the box is at 100. |
|  | layer | The layer to draw the image inside the pictures block. Known as "z-index" in HTML. By default, everything is on layer 1. However, you can make pictures appear on top of others by raising the layer number.  You can also specify a picture is "top" or "bottom".  Two pictures on the same layer generally appear in the order they were created in the XML book file.  Example:  layer="10" |
|  | zoomedLayer | An integer, starting with 1. This is the layer in the picture or grid group that the zoomed picture will go to. By default this is the top.  This means you can have a picture stacked below some others, then move it above or below them when it is tapped.  In the example, below, the lowest picture will zoom to the top (img1.jpg); the second picture will zoom to the lowest layer; the third picture appears in-between the others when zoomed.  <pictures>  <picture layer="10" zoomedLayer="top" filename="img1.jpg" scale="0.5" zoomedScale="1" />  <picture layer="20" zoomedLayer="1" filename="img2.jpg" scale="0.5" zoomedScale="1" />  <picture layer="30" zoomedLayer="25" filename="img3.jpg" scale="0.5" zoomedScale="1" />  </pictures> |
|  | restoreLayersOnZoomOut | "true" or "false".  If true, then when the picture is unzoomed — restored to its original size, rotation, and position after zooming — *all* the pictures will be restored to their original layering. For example:  You tap picture 3 of 5, and it zooms front. Next, you tap picture 2, and it zooms front. If picture 3 has "restoreLayersOnZoomOut" set to "true", then it goes back to layer 3. If not, it is behind the latest picture but above all others.  This is useful if you choose one item from a stack — of cards, for example — and you want it to return to its place.  If you have stacked items that cover each other, this setting means you *cannot* cycle through them! Instead, you will switch between the top two items! (If you tap while a transition is still happening, you might get the third item to show.)  However, a reader will probably expect an object which has moved to the top of a stack to remain there. |
|  | dimBackground | "true" or "false". If true, then when a picture is zoomed by clicking on it, a dark layer will dim the rest of the screen below it.  *Default is true.* |
|  | noThumbnail | If set to "true" then the system will not look for a thumbnail for this picture. |
|  | softEdgeMask | If "true" then use a soft-edge mask that fades into the background. |
|  | addCorners | Uses a "picture corner" graphic at each corner of the images. |
|  | frame | (integer) If this has a value, then add a frame around the image with the specified thickness. |
|  | frameStyle | Set the frame style, if frame is set (above).  Values:  “solid” : A solid stroke centered on the image edge  “thin-thick” : 25% inner stroke, 50% out, with 25% padding  “thick-thin”: 50% inner stroke, 25% out, with 25% padding |
|  | frameColor | The frame color, an RGBa color string, e.g. “40,40,50,50%” or “10,30,50,128” |
|  |  |  |
|  | singleTap | This shape zooms (or moves) with a single-tap. Set to “true” or “false”.  This setting overrides the settings.app.doubleTapExpandsPictures value, which defaults to “true”. |
|  | doubleTap | This picture zooms (or moves) with a double-tap. Set to “true” or “false”.  This setting overrides the settings.app.doubleTapExpandsPictures value, which defaults to “true”. |
|  | canPinchZoom | If true, then the reader can pinch/zoom the image even if the image cannot be tapped/double-tapped to zoom it.  Value: true or false  Default: false |
|  | pulse | The pulse attribute makes the picture pulse on screen. The value has three options: speed, amount, style.  speed : number, 1 is slow. Rate of pulsing increases as speed increases.  amount: The size of the pulse, where 1 is a small amount.  style: bounce, hbounce, vbounce all evenly increase and decrease size, starting from 100% and going up. The h and v bounces only affect horizontal and vertical, respectively.  style: wobble pulses, alternative vertical and horizontal, like water. |

|  |  |  |
| --- | --- | --- |
| <shapes> | | a list of shapes, e.g. rectangles, circles, etc. |
| attributes | x | Distance in pixels of the top-left corner of the picture block from the **center** of the screen. If x=0, the picture block top-left corner is in the middle of the screen. Or, x can be a relative position on the screen, values are : left, right, center.  “left” makes the image flush-left.  “center” centers the image on the page.  “right” makes the image flush-right.  Typically, you will use "left" to position the block on the left side. Use the x values of the <picture> elements to position them from the left. |
|  | y | Distance in pixels of the top-left corner of the picture block from the **center** of the screen. If y=0, the picture block top-left corner is in the middle of the screen. Or, y can be a relative position on the screen, values are : top, bottom, center.  Typically, you will use "top" to position the block at the top. Use the y values of the <picture> elements to position them from the top. |
|  | absolute | If absolute = "false" then the pictures block uses the textblock margins from the settings file: textblockBkgdTopMargin, etc.  If absolute="true" then the pictures block measures from 0,0 of the screen.  Default is "true"!  To get the pictures block at the top/left of the screen, set absolute="true", x="left", y="top". |
|  | top, right, left, bottom | You can set the margins for the pictures block by setting top, right, left, and bottom.  Example: top="100" |

|  |  |  |
| --- | --- | --- |
| <shape> | | a shape on the page |
| attributes | name | The type of shape:  rectangle, circle |
|  | x | pixel distance from left inside the block of shapes, e.g. "100", or "left", "right" or "center" placement in the screen. |
|  | y | pixel distance from top inside the block of shapes, e.g. "100", or "top", "bottom", or "center" placement in the screen. |
|  | width | width of the shape as it appears on screen. For circles, this is the diameter |
|  | height | height of the shape as it appears on screen |
|  | x2 | Defines second X value of a rectangle, if width not used |
|  | y2 | Defines second Y value of a rectangle, if height not used |
|  | color | A string HD color table, e.g “1, 1, 0.5, 0.2” or if you’re using the older RGBa, e.g. “250,100,100,255” |
|  | stroke | The width of the stroke around the object |
|  | strokeColor | A string HD color table, e.g “1, 1, 0.5, 0.2” or if you’re using the older RGBa, e.g. “250,100,100,255” |
|  | strokeStyle | Not sure if this works. Probably attempts for cool styles |
|  | alpha | alpha (transparency) of the shape as it appears on screen, 0.0 - 1.0  This is better done with an RGBa color setting, e.g. {1,1,1,0.5} |
|  | rotation | rotation of the shape, where 0 is not rotated, and 0 is the top of the circle. Example: "-45" would rotate the shape 45° to the left. |
|  | referencePoint | The reference point ("TopLeft", "TopCenter", "TopRight", "CenterLeft", "Center", "CenterRight", "BottomLeft", BottomCenter", "BottomRight".  ~~The shape is always anchored using the top-left. That is, the x,y refer to the top-left corner of the shape.~~ |
|  | noShadow | If noShadow="true" then there will no drop shadow behind the shape element. The default is “true,” for no shadow. |
|  | layer | The layer to draw the image inside the shapes block. Known as "z-index" in HTML. By default, everything is on layer 1. However, you can make shapes appear on top of others by raising the layer number.  You can also specify a shape is "top" or "bottom".  Two shapes on the same layer generally appear in the order they were created in the XML book file.  Example:  layer="10" |

|  |  |  |
| --- | --- | --- |
| <links> | | a list of links, which are buttons that jump to other pages |
| attributes | x | pixel distance from left for the block of links, e.g. "100" |
|  | y | pixel distance from top for the block of links, e.g. "100" |

|  |  |  |
| --- | --- | --- |
| <link> | | a link in the links list |
| attributes | id | a unique id for the link, e.g. "chapter1" |
|  | chapter | the number of the chapter to go to, in the book, counting from 1. The cover is chapter 1. |
|  | pagenumber | the number of the page in the chapter to go to. The first page of the chapter is 1. |
| (see note, below) | x | pixel distance from left inside the block of links, e.g. "100".  +x sets the distance from the previous item in the list. So, if the previous link is at 100 pixels, +30 means the link will be at 130 pixels. Useful for lists. |
| (see note, below) | y | pixel distance from left inside the block of links, e.g. "100". (see note, below)\*  +x sets the distance from the previous item in the list. So, if the previous link is at 100 pixels, +30 means the link will be at 130 pixels. Useful for lists. |
|  | imagefile | an imagefile for the button, e.g. "button1.png" |
|  | over | an imagefile for the button when the user is touching it, e.g. "button1-over.png" |
|  | linktext | text to show for the link, if there is no graphic |
|  | linkTextSize | font size in pixels of the link text |
|  | textcolor | An color string, e.g. "100,200,255" |
|  | textovercolor | An color string, e.g. "100,200,255" |
|  | font | A font name |
|  | emboss | true/false |
|  | backgroundcolor | RGB-A color of a rectangle around the link, e.g. 150,100,100,30% |

|  |  |  |
| --- | --- | --- |
| <buttons> | | a list of buttons, which are buttons that can call pre-defined functions. |
| Attributes | x | pixel distance from left for the block of links, e.g. "100", or "left", "right", ~~"screencenter",~~ "center". ~~Note that "center" uses the textblock margins from the settings.xml file.~~ |
|  | Y | pixel distance from top for the block of links, e.g. "100", or "top", "left", or "center" of the screen. |
|  | Align | Horizontal alignment of the buttons: left,right,center.  Useful when you want to align a vertical column of buttons, esp. text buttons.  *Default: center* |

|  |  |  |
| --- | --- | --- |
| <button> | | a button in the buttons list |
| attributes | id | a unique id for the button, e.g. "gotowebsite" |
|  | action | Function of the button.  ***See the action list, below.*** |
|  | pageaudioplay | If "true", then this button will be linked to the navigation bar and caption play/pause buttons. It will be *hidden* when a sound is playing, and *shown* when a sound is paused or no sound is playing. |
|  | pageaudiopause | If "true", then this button will be linked to the navigation bar and caption play/pause buttons. It will be *hidden* when no sound is playing or the current sound is paused, and *shown* when sound is playing. |
|  | condition | *See "conditions", above.*  If the condition returns true, the button will appear. Otherwise, it will not appear. |
| (see note, below) | x | pixel distance from left inside the block of links, e.g. "100".  +x sets the distance from the previous item in the list. So, if the previous button is at 100 pixels, +30 means the button will be at 130 pixels. Useful for lists. |
| (see note, below) | y | pixel distance from left inside the block of links, e.g. "100". (see note, below)\*  +y sets the distance from the previous item in the list. So, if the previous button is at 100 pixels, +30 means the button will be at 130 pixels. Useful for lists. |
|  | align | Horizontal alignment of the buttons: left,right,center.  If this is blank, the value for <buttons> will apply, if it is set. Otherwise, this value overrides the <buttons> value.  Useful when you want to align a vertical column of buttons, esp. text buttons.  *Default: center* |
|  | params | a string used for whichever function the button calls. It might be empty, a comma separated string, e.g. "red,green,blue", or simply "red", depending on the function called. |
|  | imagefile | an imagefile for the button, e.g. "button1.png".  If no "over" parameter is supplied, then the touching or "over" button file will be the image file name with "-over" added, e.g. "button1.png" uses "button1-over.png" |
|  | over | An imagefile for the button when the user is touching it, e.g. "button1-over.png".  *This is only needed to specify an 'over' file different from the automatically calculated one (see 'imagefile' above).* |
|  | width | (optional) Width of the button image to display in pixels. |
|  | height | (optional) Height of the button image to display in pixels. |
|  | sizeToFit | If "true" then the button image is resized to fit proportionally inside of the dimensions, width x height. If either height or width is missing, the image will be proporationally scaled to fit the known dimension.  *For example, if width="100" then the image will be scaled to 100 pixels wide. If the height="50" the image will fit inside of 100x50 pixels.* |
|  | hidden | If "true" then the button is initially hidden. Useful for an " pageaudiopause" button. |
|  | text | Text to use as a button, instead of a graphic image file. |
|  | font | Font name |
|  | size | Font size |
|  | textcolor | RGBA color |
|  | textOverColor | RGBA color |
|  | emboss | true/false : emboss the text |
|  | labelXOffset | x offset of label text for the button |
|  | labelYOffset | y offset of label text for the button |
|  | pulse | (see pulse in the picture definition) |

|  |  |  |
| --- | --- | --- |
| button actions |  | These are actions you can use button "action" attribute.  *Example: action="goPage"* |
|  | activatePanorama | Open a panorama on the page.  params = "pano\_id" |
|  | hidePanorama | Close a panorama on the page.  params = "pano\_id" |
|  | goPage | Jump to a page. params = "chapternumber, pagenumberinchapter" |
|  | goPageByID | Jump to a page. params = "id" of the page to go to. |
|  | sendEmail | Leaves the book to send an email with the mail app. params =me@email.com |
|  | showOverlay | show the overlay |
|  | toggleOverlay | toggle the overlay |
|  | clearTwitter | clears twitter login information (NOT IN USE) |
|  | facebookLogout | clears facebook login information |
|  | playBookAsShow | plays the book as a slideshow, starting at the showing page |
|  | goToURL | (params=url) Leaves the book app and opens the URL, e.g. "http://google.com/", in Safari |
|  | playVideo | (params=filename) Plays a video file using the system video player, a full-screen player. Good for movies.  params: filename, showControls (true/false)  *Example: params="\*/myfile.mov, true"* |
|  | playNativeVideo | The parameters, in order:  filename = params[1],  width = params[2],  height = params[3],  x = params[4],  y = params[5],  removeWhenFinished = params[6],  seek = params[7],  stroke = params[8],  dim = params[9] : opacity of dimming black background, e.g. 200 or 75%  tapToStopMsg = params[10]: Text to display telling the user to tap the screen to close the video.  Use “” to not dim the rest of the screen.  *Example: params="\*/myfile.mov, 640,480,50%,50%,true,15,1,75%,Tap to close!”* |
|  | popup | Popup an image file on screen with a white or black background.  Note the white/black background actually loads a graphic PNG file from the \_ui folder prefixd by “popup-“, e.g. “popup-white.png”.  (params = { filename, doDimScreen,white|black, backgroundAlpha, fadeTime, cancelOnTouch". Enter the param as a comma-separated string. You can leave values out.)  Example: "myfile.png,black,0.75,400,true" |
|  | popupByID | Popup an existing page object, such as a text block. The object must have it’s id set!  Unlike “popup", backgroundAlpha controls the opacity of the backdrop. And, the backdrop is not an image file. The entire screen is dimmed.  dimerams = "id, doDimScreen,white|black, backgroundAlpha, fadeTime, cancelOnTouch". Enter the param as a comma-separated string. You can leave values out.)  Example: "text1,true,white,400,true" |
|  | webPopup | (params = "targetURL, [white|black], bkgdAlpha(0-1), fadeTime (in milliseconds), internet required [true|false]").  The internet required flag defaults to true.  Example:  "http://mysite.com/,white,0.75,400, true" |
|  | playAudio | (params=filename, hideMessages) Plays an audio file. If you include "hideMessages", then there will be no message on screen saying "Play" or "Pause". |
|  | pauseAudio | (params=filename) Pauses a playing audio file |
|  | playPauseAudio | (params=filename) Plays and pauses an audio file, with alternate taps. |
|  | purchaseItem | (params=id) Starts in-app purchase of the object with the specified ID. |
|  | autoplayAudioOn | Turn on autoplay audio. Autoplay audio will start normally. |
|  | autoplayAudioOff | Turn off autoplay audio. Audio will not play automatically even on pages with autoplay audio on them. |
|  | showCaption |  |
|  | toggleCaption |  |
|  | hideCaption |  |
|  | tellPlugin | Sends information to a plugin. This is used to make a plugin do something, such as start a game.  The ‘params’ attribute should be a key=value pairs, joined with commas, e.g. “id=mypluginname,action=shrink,height=10,id=100,target=cow”.  If there is an ID provided, the value of the ID will determine which plugin gets the values. If the ID matches the chapter plugin ID, these values will be send to the chapter plugin. If the ID matches a page plugin, the values go there.  If no ID is provided, the default target is the page plugin. |

|  |  |  |
| --- | --- | --- |
| <sprites> | | The set of sprites on a page |
| attributes | x | Distance in pixels of the top-left corner from the top-left of the screen. Or, x can be a relative position on the screen, values are : left, right, center.  Typically, you will use "left" to position the block on the left side.  Default is 0. |
|  | y | Distance in pixels of the top-left corner of the shape block. If y=0, the shape block top-left corner is in the middle of the screen. Or, y can be a relative position on the screen, values are : top, bottom, center.  Typically, you will use "top" to position the block at the top. |
|  | absolute | If absolute = "false" then the shapes block uses the textblock margins from the settings file: textblockBkgdTopMargin, etc.  If absolute="true" then the shapes block measures from 0,0 of the screen.  Default is "true"!  To get the shapes block at the top/left of the screen, set absolute="true", x="left", y="top". |
|  | autoplay | true/false: if true, the sprite starts playing as soon as the page is opened. |
|  | filename | Path and filename of the sprite sheet, e.g. \_user/mysprites/greenman.png. |
|  | x | Initial x position of the sprite, relative to the <sprites> positioning. |
|  | y | Initial y position of the sprite, relative to the <sprites> positioning. |
|  | width | Width of a single sprite in the sprite sheet, e.g. 128 |
|  | height | Height of a single sprite in the sprite sheet, e.g. 128 |
|  | loop | Looping of the sprite animation:  0 = loop forever  1 = play once  n = play n times |
|  | rotation | Rotation of the sprite in degrees |
|  | framecount | Number of frames on the sprite sheet |
|  | start | For consecutive-frame sequences, start represents the starting frame, and count represents how many frames from the start that the sequence should end. |
|  | count | For consecutive-frame sequences, start represents the starting frame, and count represents how many frames from the start that the sequence should end. |
|  | time | This is the time (in milliseconds) between each frame. If this is not specified, then animation will be based on the framerate of your app. |
|  | loopdirection | This can either be “forward” or “bounce”, with “forward” being the default if don’t set this parameter. The “bounce” option will play forward, then backwards through the sequence of frames. |
|  | pathRotationOffset | Correction to rotation for motion paths. If your sprite appears sideways when you add motion, you'll probably have to correct it, probably by adding 90 here. |
|  | doRotateOnPath | If you do NOT want your sprite to rotate as it moves along a motion path, set this to "false". |
| elements |  |  |
|  | path | A set of x,y coordinates the define a motion path. The path will be smoothed, so only add the necessary points. The path is written as a list of numbers, like this:  10,10,20,20,30,30  where each pair is an x,y coordinate. So, the above will be read in pairs, like this:  10,10 20,20 30,30  \* You must have at least three points! |

|  |  |  |
| --- | --- | --- |
| <animation> | | A looping animation based on individual image files. It goes really fast, probably 30 fps, and I don't know how to slow it down. |
|  | x | x position on screen |
|  | y | y position on screen |
|  | basefilename | base name of the series of images, e.g. "myshow". This assumes each image will be named with the extension plus a zero-padded number. For example, myshow-01.png, myshow-02.png, etc. Pad only as necessary. If there are 10-99 images, the numbers start 01 and go to 99. Do not use 001 if there are less than 100 images. |
|  | extension | extension of the filenames, e.g. "png" or "jpg" |
|  | framecount | number of images in the animation. |

* Currently there is room for only one plugin. This will change soon.
* Plugins should have a \_removeSelf() function if they need to do extra tasks, such as removing Runtime handlers, before the group itself is removed. The \_removeSelf() built into the plugin group MUST remove the group! That's because we can't do it from above, since the custom tasks might take some time.

|  |  |  |
| --- | --- | --- |
| <plugin> | | Use a Corona display object created by a module. |
|  | x | x position of the plugin on the screen, using pixels from the top-left or left/right/center or a percentage of the screen, from the top-left. |
|  | y | y position on the table of contents, either numbers or a percentage of the screen height |
| elements | <filename /> | The name of the plugin module. It must reside in the main directory of the app. Do not add the ".lua" suffix. For example, if your plugin is "memorymatch.lua", then filename = "memorymatch" |
|  | All other elements | All other elements of the plugin element are parameters for the plugin. |
| *Example:*  <plugin x="center" y="center">  <filename>memorymatch</filename>  <scale>0.8</scale>  <hspace>280</hspace>  <vspace>130</vspace>  <rowpadding>20</rowpadding>  <width>800</width>  <height>600</height>  <top>70</top>  <matchTextY>170</matchTextY>  <orientation>"landscape"</orientation>  <maxSoundDuration>1000</maxSoundDuration>  <localpath>\_user/game/</localpath>  </plugin> | | |

|  |  |  |
| --- | --- | --- |
| ~~<tableofcontents>~~  (See “contents” attribute of <page> element  and  “isContents” attribute of <text> element) | | ~~Insert an automatically generated table of contents for the book, with links.~~  ~~A chapter must have a <title> element to be included. Otherwise, we wouldn't know what to call it.~~  ~~If a chapter has the attribute, notInContents="true", then it will not appear in the table of contents.~~  ~~Example:~~  ~~<tableofcontents x="25%" y="25%" title="My Contents" absolute="true" />~~ |
|  | x | x position on the table of contents, either numbers or a percentage of the screen width |
|  | y | y position on the table of contents, either numbers or a percentage of the screen height |
|  | ~~title~~ | ~~Title text for the table of contents~~ |
|  | absolute | If absolute is "true", then do *not* use the margin settings for a textblock around the contents. If "false", then use the textblock margin settings. |
|  | shadow | Add a drop-shadow to the contents entries. |

|  |  |  |
| --- | --- | --- |
| <panoramas> | | Block of 360° panoramas for a page.  Panoramas should be called by a button since they fill the screen and have a "close" button. |
|  | x | x position on the screen. |
|  | y | y position on the screen. |
|  | absolute | If absolute is "true", then do *not* use the margin settings for a textblock around the contents. If "false", then use the textblock margin settings. |

|  |  |  |
| --- | --- | --- |
| <panorama> | | A 360° panorama entry in the paroramas block. |
|  |  |  |
|  | filename | The **base** filename for the four images that make the 360° panorama. If the base name is "mypano.jpg", then you must have "mypano-1.jpg", "mypano-2.jpg", "mypano-3.jpg", and "mypano-4.jpg".  The files *must* be the same dimensions, e.g. 1024x768 pixels. |
|  | multifile | True/False: whether the image is split into four files or is just one file.  *Default is true.* |
|  | panoOverlay | A display object, e.g. a group or image. This is useful for creating a layer with buttons, text, and graphics to show over the background image. |
|  | backgroundcolor | An RGBA color in string form, e.g. "128, 128, 128, 100%"  Note that the alpha can be either a percent or integer value (0-255)  *Default is black, 0, 0, 0, ,100%* |
|  | orientation | "landscape" or "portrait", this determines which way the panorama is stiched together, vertically or horizontally from the four files.  *\*Optional if "multifile" is false.*  *Default value is "landscape".* |
|  | noWrap | If true then, the panorama will not wrap around when the viewer hits the end. It will stop. Useful for a less than 360° view.  *Default is false.* |
|  | x |  |
|  | y |  |
|  | zoom |  |
|  | tiltAngle |  |
|  | imageScaleX |  |
|  | imageScaleY |  |
|  | touchonly |  |
|  | navtype |  |
|  | maxDistanceToSlide |  |
|  | activateTime |  |
|  | showCloseButton |  |
|  | closeButtonDefault | Path to file of the close button, e.g. "\*/\_user/btn/close.png" |
|  | closeButtonOver |  |
|  | closeButtonX |  |
|  | closeButtonY |  |
|  | overlay |  |
|  | overlayX |  |
|  | overlayY |  |
|  | margins |  |
|  | absolute |  |
|  | buttons | Overlay buttons that moves with the image. Create just like regular buttons:  <buttons> <button>…</button>  </buttons> |

*\* An important note on how to position links and buttons on the screen:*

The smallest x of the <link> elements is leftmost element in the links. For weird technical reasons, <link> elements always start at (0,0) inside the <link>. So, if the link 1 is at (10,10) and link 2 is at (20,20), then link 1 will show at (0,0) and link 2 at at (10,10), *inside the <link>.* The system throws away the extra 10 pixels.

Actually, this makes your life easier! All you need to do is set the (x,y) of the <link> to the leftmost element on the screen:

You can simply take the (x,y) of each slice from your Photoshop screen design file (double click the slice to see the x,y). Be sure to set the (x,y) of the <link> to the top-left corner of a box holding all your slices. You don't need to do any math!

### Plugins

The system comes with some built-in "plugins." These are specialized games and such. You can use only one plugin on a page.

#### drag-and-drop

The "drag-and-drop" plugin lets you make any picture on the page into a drop target or a draggable item (to be dropped).

First, add the plugin to the page, as <plugin> element. Here is an example:

|  |
| --- |
| <plugin x="center" y="center">  <filename>scripts.drag-and-drop.game\_engine</filename>  <localpath>scripts/drag-and-game/</localpath>  <testing>false</testing>  <startColor value="255,100,100,255" />  <enterColor value="255,0,0,255" />  </plugin> |

The elements are:

|  |  |
| --- | --- |
| filename | Path to the plugin in the scripts folder, using periods instead of slashes. |
| localpath | Path to the plugin, using slashes. |
| testing | Used for debugging code. |
| startColor | A tint for a drop target when the user picks up something draggable.  Use a transparent color, e.g. "0,0,0,0" to have no color change. |
| enterColor | A tint for a drop target when the user hovers over the drop target while 'holding' a draggable item.  Use a transparent color, e.g. "0,0,0,0" to have no color change. |

You can add these attributes to a picture element:

|  |  |
| --- | --- |
| DROP TARGETS |  |
| droptarget | true/false  If true, the picture element is something you can drag things onto. Example:  <picture droptarget="true" ... /> |
| dropgroup | The drop groups must match in the targets and draggable items.  So, this is the name of the kind of thing you can drop into a drop target. For example, if you tag your draggable items "fish" and the dropgroup is "fish", then the drop target will accept it. If the draggable were "mice" then it a drop target with dropgroup "fish" would *not* accept it.  Example:  <picture ... dropgroup="fish" droptarget="true" ... /> |
| action | An action to perform if the dragged item is accepted by a drop box. These are the same actions you can do with a button (see above).  This uses the newer action function, so now you can **chain actions,** depending on the action!  If the action allows chaining, such as "", Use the params |
| params | Parameters for the action, same as with a button (see above).  Example:  <picture ... action="playAudio" params="\*/audio/seal.mp3,hideMessages" /> |

|  |  |
| --- | --- |
| ACTIONS | (see button actions, above, for most actions)  **Chaining Actions:**  An action to perform if the dragged item is accepted by a drop box. These are the same actions you can do with a button (see above).  This uses the newer action function, so now you can **chain actions,** depending on the action! Chainging simply means you can have actions follow each other, like dropping an object followed by playing a sound.  If the action allows chaining, such as "dropObject", use additonal params to spec the new action. The first additional param is the next action name, e.g. "playAudio", and the following params are whatever that action needs, e.g. "mysound.mp3,hideMessages"  Example:  <picture ... action="dropObject" params="move,playAudio, \*/audio/seal.mp3, hideMessages" /> |
| dropObject (chainable) | (params="move|hide")  If a screen object is dropped and accepted, it can be moved to the drop point, or hidden.  Example:  <picture ... action="dropObject" params="move" ... /> |

|  |  |
| --- | --- |
| DRAG OBJECTS |  |
| draggable | true/false  If true, the picture element can be dragged onto a drop target.  <picture draggable="true" ... /> |
| dropgroup | **In a drag object:**  The drop groups must match in the targets and draggable items.  So, this is the name of the kind of thing you can drop into a drop target. For example, if you tag your draggable items "fish" and the dropgroup is "fish", then the drop target will accept it. If the draggable were "mice" then it a drop target with dropgroup "fish" would *not* accept it.  Example:  <picture ... dropgroup="fish" draggable="true" ... /> |
| singleUse | true/false  If this is true, then the dragged object will NOT be draggable once it has been successfully used, i.e. the drop target accepted it. |
| dragWidth, dragHeight | The object to drag is represented by a rectangle the same width and height as the picture or object that is marked draggable. However, you may wish a larger area to touch and drag, especially with small objects. Set the dragWidth/dragHeight to do this.  <picture ... dragWidth=”100” dragHeight=”100” ... /> |

#### memory-match

## Templates

The templates file is build almost exactly like a book file. See the example.

Special codes:

|  |  |
| --- | --- |
| {null} | An element containing the {null} code will always be empty, even if the page of the book file has a value. For example, if the book file has  <imagefile>myvalue</imagefile>  but the template has  <imagefile>{null}</imagefile>  then the final page value will be nothing at all.  This is useful for making sure a book value does not appear when you don’t want it, while allowing you to have all kinds of values in your book file, typically what would happen from a good data dump. |
| {element name} | If you include an element name with brackets, e.g.  {date}  inside another element value in a template, e.g.  <caption>The date was {date}.</caption>  Then the value of the element named will be inserted. For example:  The book file contains:  <date>12/5/2005</date>  and the template contains:  <date>The date was {date}.</date>  <caption>On {date} it happened.</caption>  the result will be a caption reading,  On 12/5/2005 it happened. |

## PAGE METADATA UPDATE FILES

In addition to the page data in the book.xml file, you can have an update XML file, called page\_metadata.xml, which overwrites values in the book.xml file.

The purpose of this update file is to allow you to update page values, such as captions, without having to modify your book file. If you are developing a book, you can build the book while allowing for changes to captions and such without having to worry about updating your book file.

Since the update XML file is simple XML format, really just a list of key/value pairs wrapped in <items></items>, it is easy to maintain as a database (or generate from Adobe Lightroom). A client could be correcting items in this list even as your are building out the book.

The key is to be sure pages have unique ID values, i.e. <page id="myid">. The ID of a page in the book is matched to the ID of a page in the update file.

Hint: Since this updating idea was designed to deal with books of images — with one main image per page — the easiest way to have unique page ID is to use the filename of the image file for the page ID, on the assumption you won't be repeating images. When you export your XML code for your book, *and* when you make the XML code for the update file, be sure your

The XML file looks like this:

|  |
| --- |
| <items>  <page id="X">  <name>value</name>  <name>value</name>  <name>value</name>  …  </page>  …  </items> |

## InDesign Bookmaker Plugin

The script for InDesign that creates a book for Photobook works by parsing the InDesign document, and creating an XML structure that Photobook can understand. InDesign allows for XML structures inside of a document, and that's where the XML is contained. The user must export the XML from InDesign to use it. The script scans the InDesign file for page items, such as images and text boxes. It creates XML items based on these.

Because Photobook and Corona cannot handle character styles (e.g. bold, or color changes), there are definite limitations to how well we can get text to work.

The current script was created as needed to finish the Russian California book, so its abilities may seem a little random. The script can create (with some success) :

1. text boxes
2. text paragraph styles
3. images
4. basic buttons that go to pages
5. basic image animations
6. rectangles and basic object frames with three kinds of borders
7. text links to pages and anchors: These are problematic because the system creates both a text block and a hyperlink, and not always in exactly the same place. I don't have a way to handle hyperlinks inside of text, which means hand-fixing of the actual XML code.

Some major features not implemented include:

1. Full implementation of overlays — currently they are only images
2. Audio and video
3. Plugins
4. Customization of the navigation bar
5. Table of Contents creation
6. Image layers, esp. changing layers when "zooming" and image.
7. Selection of which page pictures for uploading to Facebook
8. Book introduction (video, settings)
9. Inventory and In-App purchasing

Also, the Photobook settings file is independent of the InDesign, but one could rewrite some of it based on the InDesign.

The system cannot properly handle multi-chapter documents, so chapters have to be created by hand, one at a time, from individual InDesign files. Then, you have to stitch it all together into a book.

### How to install

To install the InDesign plugin, put the Javascript into the user Scripts folder in InDesign. To do this,

1. show the "Scripts" window in Indesign (found under Windows:Utilities:Scripts menu).
2. right-click the "User" folder, and "Reveal in Finder". You will see a folder containing "Scripts Panel".
3. copy the provided javascript, called "Tag XML for Photobook.jsx", into the "Scripts Panel" folder.
4. Also copy the file "Pasteboard to 200x200.jsx". It is useful when the script crashes — it resets the InDesign pasteboard to something reasonable.

### Initial InDesign File Setup

1. Create an InDesign document at pixel dimensions of your book, e.g. 1024x768. (Don't worry about retina (double-resolution) sizes. For a retina iPad, which can show hires images at 2048x1536 pixels, you will still use the normal sizing.)
2. You might wish to use Facing Pages for layout, but you must be careful. Graphics that you see spill over two pages won't do so in a book — they will only be on one page. You will have to split them into two matching parts and put part on each page.

### Preparation of an Existing InDesign File

1. Open InDesign file.
2. Choose the File:File Info menu. Set the book title and author.
3. Open the Scripts window, and double-click (or run) the "Tag XML for Photobook" script.
4. Check the bottom-most box, "Set up the document…"

* ~~You will have to choose the "DID" file, which is in the Photobook Publishing Tools folder in Dropbox.~~ The "DID" file will be automatically chosen for you.
* Choose a folder in which to save your book and/or template.

### Using the Script

* The setup will create some layers for you: PB\_Overlays, PB\_Captions, PB\_Background.
  + A picture on the Overlays layer will become the overlay picture
  + The first text box on the captions layer is the caption text.
  + Images put onto the PB\_Background layer will not be included on the page, but will appear on the background file for the page. I think….
* \* To create a page as JPG file, mark the page with the color label, "Red".
* There are two picture settings:
  + To make an image zoom to center when tapped, set its Object Style to "PB Zoom To Center"
  + To make an image zoom its uncropped version to center, set its Object Style to "PB Zoom Original"
* Text blocks should have "Text Frame Options: Baseline Options" set to "First Baseline: Ascent". I think… Even then, it ain't perfect.
* Text Formatting: the format of a paragraph follows the 1st character. In general do NOT use formatting beyond the Paragraph Style. If you set the first character to ALL CAPS, for example, the whole paragraph will be all caps.
* Chapter Numbering: Be sure to choose the menu item "Layout:Numbering & Section Options…" if you plan to have more than one chapter in your book.
* *Table of Contents:* A text block can hold a table of contents. Choose the object style "pb Contents" for that text block.

### Field Names

You can make label text blocks so that WordPress knows where to put custom fields. For example, you might have a custom field, "scientific-name", and you want that in a special text field in the template.

Do indicate that a text block gets data from a custom field, type anywhere in the text block your field name surrounded with double-curly brackets, like this:

{{scientific-name}}

To export the book XML, choose File:Export in InDesign. Export "XML" format. Turn off ALL the checkboxes in both panels. Export "book.xml" into the \_user folder to have it appear in Corona.

You can also export pictures, overlay pictures, page backgrounds, etc. You won't want to do this every time — it's slow, and the Corona simulator keeps trying to redraw with every file change — so these functions are separated in the plugin panel.

### Sample document

I have provided a simple, sample document. It doesn't show every possibility, but it mostly works.

* Major Issues
* Corona doesn't handle fonts properly, so vertical positioning is a constant problem. Also, horizontal font spacing is different between InDesign and Corona, so text doesn't always match up. Very annoying. The modules in Photobook, "fontmetrics.lua" is designed to improve this, but is definitely imperfect. I'd like to get some font-reading functionality into Photobook, but that's a slog!
* The lack of character-based formatted in iOS is also a major limitation, only solved by improving my text engine in Photobook.

# Developing Plugin Scripts

Books can use standardized “plugin” Lua modules *that are compiled into the original app.* These are not true plugins — they cannot be added after the app has been compiled and published — but they do allow a developer to extend the possibilities of an app without tinkering with the core code.

These plugins can be executed from settings in the XML code.

The viewer software runs chapter plugins when a chapter is loaded. Such a plugin includes the “swim” plugin which makes creatures swim on pages in a chapter.

More common are page-based plugins, often games.

A page plugin is loaded when the page is created by the viewer software, usually a few pages before where the user is looking at the app.

Each plugin is sent parameters when it is first called, when the viewer loads the page it is embedded into. These include all XML elements in the <plugin> element, *and* information about the current XML book and the app.

Book and system information includes:

-- The path to the book (useful for getting assets)

params.bookpath = book.path

-- The system directory, e.g. system.CachesDirectory

-- (useful for getting assets)

params.bookSourceDirectory = book.sourceDirectory

-- Pass page structure to the plugin

params.pageValues = pageValues

-- Must be careful to get rid of these to avoid memory leaks!

params.pageScreen = pageScreen

-- Pass on slideview handlers from whatever called it:

-- showFloatingBook is used by the cardviewer, for example

params.handlers = handlers

-- These are built-in button handlers, things we could use for fun!

params.bookhandlers = handleUserAction

-- Pass all system settings

params.settings = systemSettings