



Scripture
App Builder

Building Apps



Scripture App Builder: Building Apps

© 2023, SIL International

Last updated: 21 March 2024

You are free to print this manual for personal use and for training workshops.

The latest version is available at
<http://software.sil.org/scriptureappbuilder/resources/>

and on the Help menu of Scripture App Builder.

Contents

1. Preparing content for your app	7
1.1. Preparing text	7
1.2. Preparing images	9
1.3. Preparing audio.....	9
2. How to build your first app	10
3. Installing the app on your phone	13
4. App Creation Basics	17
4.1. How should I choose the app package name?	17
4.2. Where do the default book names and abbreviations come from?	18
4.3. Do I have to create a new keystore for each app, or can I reuse the same keystore for several of my apps?	19
4.4. How can I include pictures in my apps?	19
4.5. How can I add a horizontal line to separate certain paragraphs in text?	20
4.6. I don't like the name "Scripture App". Have you thought of calling the app something else?.....	20
4.7. Can I build apps when I do not have internet access?	20
4.8. How do the glossary entry popups work?	21
4.9. Can I build an app from the command line?	22
5. Fonts	24
5.1. What is Grandroid?.....	24
5.2. When do I need to include the Grandroid libraries?.....	25
5.3. What is GeckoView?	25
6. Audio.....	27
6.1. How can I associate audio files with the text?	27
6.2. How do I create the audio timing files for audio-text synchronization?.....	28
6.3. How do I distribute the audio files with the app?	28
6.4. How are the timing files distributed for the app?	33
6.5. How can I use audio clips in the app?.....	33
6.6. How can I create subtitles from text and timing files?.....	34
7. Video.....	35
7.1. How do I include videos in the app?.....	35

7.2.	How do I reduce the size of a video that I want to package within the app?	35
7.3.	How do I specify where the videos will be displayed within the app?.....	36
7.4.	How can I create slideshow videos using audio and timing files?.....	36
8.	About Page	37
8.1.	What information should I include in the About page?	37
8.2.	Which formatting codes can I use in the About page?	37
8.3.	Which variables can I use in the About page?.....	38
9.	Footers	39
10.	Multiple Translations	40
10.1.	Three different layouts	40
10.2.	How to add a translation to the app	42
10.3.	How to configure the Layouts dialog.....	43
11.	Navigation Drawer.....	44
12.	Contents Menu	45
12.1.	How do I create a contents menu?.....	45
12.2.	How do I create additional contents menu screens?	46
13.	Verse on Image	47
13.1.	The Verse on Image editor in the app	47
13.2.	Configuring Verse on Image in the app builder	49
14.	Verse of the Day	50
14.1.	Verse of the Day notifications	50
14.2.	Configuring Verse of the Day in the app builder	50
15.	Highlighting, Notes and Bookmarks.....	51
15.1.	Highlighting.....	51
15.2.	Notes.....	52
15.3.	Bookmarks	53
15.4.	Configuring highlights, notes and bookmarks	54
15.5.	Questions about highlights, notes and bookmarks	54
16.	Borders.....	55
16.1.	How do I add a border to the app?	55
16.2.	How does the app draw the border image?.....	57
17.	Bible Plans	57

17.1. How do I define a plan?	57
17.2. How do I add a plan to the app?.....	58
18. Book Tabs.....	60
19. Radio.....	61
20. Picture Story Books.....	62
20.1. How do I define a picture story book?	63
20.2. What do picture story books in Word documents look like?	63
20.3. What do picture story books in SFM format text files look like?	65
20.4. Where do the pictures go?	66
20.5. How can I get the pictures to move when the audio is playing?	67
20.6. What about font and font size?	67
20.7. What audio timing labels are used?	67
20.8. How can I add background music when the audio is playing?.....	68
20.9. Where can I get sample story book text and pictures?	69
20.10. How can I record the audio files?	69
21. Song Books	69
21.1. How do I define a song book?	69
21.2. What do song books in Word documents look like?	69
21.3. What do song books in SFM format text files look like?	72
21.4. How can we associate audio with each song?	74
22. Sharing Apps.....	75
23. Deep Linking	75
23.1. Setting up Deep Linking	76
23.2. Creating Deep Links	77
23.3. Deferred Deep Linking	78
24. Analytics.....	79
24.1. Firebase Analytics	80
24.2. Amplitude Analytics	81
24.3. FCBH Digest Analytics	82
24.4. S3 Digest Analytics	83
24.5. Data Payload for FCBH and S3 Digest Analytics	84
25. Push Notifications.....	85
25.1. Setting up Firebase Cloud Messaging	85

25.2. Sending messages.....	85
26. User Accounts.....	86
26.1. Setting up User Accounts in Scripture App Builder	86
26.2. Setting up the database in the Google Firebase console	86
27. Registration Screen	88
27.1. Setting up the Registration Screen in Scripture App Builder.....	88
27.2. Setting up the database in the Google Firebase console	89
28. EPUB	93
29. Progressive Web App.....	95
29.1. Why Would I Want a Progressive Web App?	95
29.2. Which Progressive Web App Should I Use?.....	95
29.3. Audio in PWA	95
29.4. Building a Static PWA.....	96
29.4.1. Installing Prerequisites for Static PWA on Windows	96
29.4.2. Installing Prerequisites for Static PWA on Linux	97
29.4.3. How do I export Static PWA files?.....	98
29.4.4. How do I Add Audio to my Static PWA?	98
29.5. Building a Modern PWA.....	98
29.5.1. Configuration for Modern PWA	98
29.5.2. How Do I Build a Modern PWA?	99
29.5.3. How Do I Test a Modern PWA?.....	99
29.5.4. How Do I Publish a Modern PWA?	99
30. Publishing and Distribution	100

1. Preparing content for your app

Before you build an app with Scripture App Builder (SAB), you need to get your content (text, images and audio) into formats that SAB can handle.

1.1. Preparing text

The text format you need depends on the type of app content you have.

- **Books of the Bible:**

The Scripture text needs to be in one of the following formats:

- Paratext files (.usfm, .sfm, .ptx, etc.)
- USX files (.usx)
- Digital Bible Library Text Release Bundles (.zip)

- **Other types of app content such as commentaries, picture story books, song books and Bible studies:**

The text needs to be in one of the following formats:

- Word documents (.docx)
- SFM text files (.sfm)
- HTML files (.html)
- Bloom books (.bloompub)

Here are some more details about each of these formats:

1. Paratext files for Bible books

SAB can import text from Paratext USFM (Unified Standard Format Marker) files. These are the files used in Paratext projects – one per book of the Bible.

```
\c 5
\s Kile Kafila yere lemu l'a pye faabobonjo ki junjo ni ge
\p
\v 1 Ayiwa, ba Yesu ya sipyiire ti na we, na dugi tiin faabobonjo ki
junjo ni. A wu wo kalaapiire t'i fulu wu na.
\v 2 A wu jo kon na pu kalaa na:
\s Piike p'i je duba nagoo pu we?
\r (Luka 6:20-23)
\q1
\v 3 «Piimu p'a li ce na funmo fee pee je
\q2 Kile shizhaa na ge,
\q2 pee je duba nagoo,
\q1 bani pee wuuro ti je
\q2 fugba saanra te.
```

In USFM files, the chapters, section headings and paragraphs are marked by standard format markers such as \c, \s and \p. For more details, please see <http://paratext.org/about/usfm>.

It is recommended that you run the Paratext Basic Checks to ensure the text is as clean as possible for publishing. Select **Checking** ➤ **Run Basic Checks...** from the Paratext main menu.

Critical checks:

- **Chapter/Verse Numbers:** Checks for duplicate or invalid chapter or verse numbers. Checks that all chapters and verses defined in the versification for the project are present.
- **Markers:** Checks you are using valid USFM markers and they are not in unexpected places.
- **References:** Checks that Scripture references, e.g. “Mat 6:1-5”, follow exact abbreviations and syntax. If not, they will not work as links in the app.

Important checks:

- All of the remaining basic checks: Characters, Punctuation, Capitalization, Repeated Words, Unmatched Pairs of Punctuation, Quotations, Numbers.
- **Spelling:** In Paratext, select **Checking** ➤ **Spell Check Current Book...**

2. Other SFM text files

SAB can import text from SFM files which contain content other than Bible books, such as commentaries, Bible studies, stories or songs. Like Paratext USFM files, the chapters, section headings and paragraphs need to be marked by standard format markers such as \c, \s and \p.

The first marker in the file must be \id XYZ, where XYZ is a code you choose. Do not choose a code already reserved for a Scripture book, e.g. GEN.

SFM book files must be **plain text files**. If you have Unicode characters, the text files should use UTF-8 encoding.

To create a text file in Windows, use a text editor such as Notepad.

To create a text file on a Mac, use TextEdit, remembering to choose Plain text files in the preferences because otherwise the default file type is RTF (which contains a lot of additional formatting codes).

For more details about how to define picture story books and song books using SFM text files, please see sections 20 and 21.

3. Word documents (.docx)

SAB can import text and images from Microsoft Word (.docx) documents.

When Word documents are displayed in the app, basic formatting will be preserved such as character styles (bold, italic, underline), numbered lists, bullet points, hyperlinks, footnotes and simple tables.

To define separate chapters or pages, insert page breaks using CTRL+Enter.

For more details about how to define picture story books and song books using Word documents, please see sections 20 and 21.

4. Digital Bible Library Text Release Bundles

SAB can import text from Digital Bible Library (DBL) text release bundles. These are zip files containing USX files for each book as well as other configuration information.

For more information about the DBL, see:

<https://www.thedigitalbiblelibrary.org/>.

You can find some DBL text release bundles available for free use on Biblica's Open.Bible website: <https://open.bible/bibles/>

5. HTML files (.html)

SAB can display HTML files and make use of associated stylesheets (CSS) and images. Audio-text synchronization is not supported yet with this format.

6. Bloom books (.bloompub)

SAB can display Bloom books, by using an embedded Bloom Player, as used in the Bloom Reader app. Books need to be in the .bloompub format.

For more information about Bloom, please see: <https://bloomlibrary.org>

1.2. Preparing images

Images are imported from Word documents or need to be specified separately if you use SFM format. They should be in JPEG or PNG format.

Keep the image size small enough so that they display well on a small screen and will not make the app size too large. If your pictures are in a Word document, you can use the **Compress Pictures** tool in Word. Otherwise, SAB will allow you to resize the images after you have added them to the app project.

1.3. Preparing audio

If you want to include audio files in your app, these need to be in MP3, 3GP or WebM (Ogg Opus) audio format. Normally this should be one audio file per page or chapter.

You can also include audio clips which are short audio files that are played when the user taps on a word, phrase or image.

If you have a picture story book, you can have a single audio file for the whole book or one audio file per page.

Keep the audio files at a size where the quality is good enough for a phone and where the file size is not too large.

2. How to build your first app

To build your first app with Scripture App Builder:

1. Launch **Scripture App Builder** from its icon on the desktop.
2. Click **New App** on the toolbar. The New App wizard will appear.
3. On the first page of the wizard, specify the **App Name**, such as “Dogon Bible”, “Livre de Luc”, “Supyiré NT”, etc. This is the main title of your app and will be seen by the user. Do not include underscores or hard to understand abbreviations.

Click **Next** to move to the next page.

4. On the second page of the wizard, specify the **Package Name**, a dot-separated string which uniquely identifies your app.

More details about choosing a good package name can be found in section 4.1 *How should I choose the package name?*

Click **Next** to move to the next page.

5. On the page of the wizard titled **Books**, click **Add Books...** and select the books you want to see in the app. These can be USFM (Paratext) files, USX files, Digital Bible Library text release bundles, Word documents (docx) or a zipped file of USFM or USX files.

Click **Next** to move to the next page.

6. If you have selected Scripture books, the next page will be titled **Book Order**. Choose the book ordering according to the church tradition of the target users.

Click **Next** to move to the next page.

7. On the page of the wizard titled **Language**, specify the language code and name of the main language used in the app. Click **Select...** to choose the language code from a list of languages. If required, select the checkbox to specify additional identification information, such as the script, region and variant.

Click **Next** to move to the next page.

8. On the page of the wizard titled **Copyright and Licensing**, specify the copyright and licensing information that you would like to appear on the About box in the app. This includes the copyright owner for the text. Use the **Copyright Helper** wizard to help you.

If you do not know what to put here, please ask the publishing department of your organisation for advice. They will want to make sure you get this right and do not simply make a guess as to what to include.

Click **Next** to move to the next page.

9. On the page of the wizard titled **Font**, choose the font. You can either select from the given list of fonts or specify a different TrueType font file.

Click **Next** to move to the next page.

10. If this page is titled **Font Choice**, choose some more fonts to add to the app. The app user will be able to change the default font to one of these fonts (in the font chooser and in the Verse on Image editor).

Click **Next** to move to the next page.

11. On the page of the wizard titled **Font Handling**, you can select Grandroid and/or GeckoView if you know that the standard Android components will have trouble displaying the text correctly (e.g. if it is a complex script). More information on Grandroid and GeckoView can be found in the **Fonts** section of this document.

Click **Next** to move to the next page.

12. On the page of the wizard titled **Color Scheme**, choose the color scheme for the app. The color you choose is the one that will be used for the main app bar. Individual colors for text, titles, links, backgrounds, etc. can be customised later.

Click **Next** to move to the next page.

13. On the page of the wizard titled **Default Interface Language**, choose the language you want users to see when they first enter the app. This can be the current system language.

Click **Next** to move to the next page.

14. On the page of the wizard titled **Interface Languages**, choose the app interface languages that the user can choose between.

Click **Next** to move to the next page.

15. On the page of the wizard titled **Features**, choose the features you want to enable in the app.

Click **Next** to move to the next page.

16. On the page of the wizard titled **Icon**, choose the application launcher icon. You can select one of the images in the table or if you have your own PNG image files for the icon, click **Browse** and select them.

Click **Next** to move to the next page.

17. On the page of the wizard titled **Signing**, you need to specify the keystore and alias to use to sign the app. An app must be signed in this way so that it can be installed on an Android device.

If you do not already have a keystore file (which you are unlikely to have if this is your first time using the program):

- i. Click **Create KeyStore**.
- ii. Enter a new filename for the keystore, such as “keystore1” or something like that. Specify a password. Click **Next** to continue.
- iii. Enter an alias name for a key to create within your new keystore, such as “key”. Specify a password, which can be the same as the password you entered on the previous page. Click **Next** to continue.
- iv. On the **Certificate Issuer** page, provide details of your organisation in at least one of the fields. Click **Next** to continue.
- v. A new keystore will be created for you. Click **Close**.

18. Back on the **Signing** page of the New App wizard, you need to specify the keystore password, select the alias and enter the alias password (just as you entered them in the step above).

Click **Next** to continue.

19. On the page of the wizard titled **Project**, you can enter modify the project name and add an optional description of the app project. Neither of these will be visible to the user of your app. They are just for your own use and might help you distinguish between multiple app projects.

Click **Finish** to continue. The New App wizard will close, and the app definition will be added to the tree view on the left of the screen.

20. Take a look at each of the app configuration pages by selecting them in the tree view on the left. Look in each of the tabs on each page to verify that you have

the settings you want. You can always go back to them later to change them if you find you need to make modifications to fonts, colors, styles, etc.

21. When you have finished configuring the app, click the **Build Android App** button on the toolbar at the top of the screen.

If something is not configured correctly for the build to work, you will be notified.

22. A compilation window will appear. Wait about a minute while the app is compiled.

The first time the build process is run, the compiler needs to connect to the internet to download some files. After this, subsequent app builds will not require internet access. See **Tools** ➤ **Settings...** ➤ **Build Settings** to turn on offline mode after the first app build.

23. If the build succeeds, you will have a new APK file – the installation file for an Android app.

The next section describes how to copy this APK file to your phone and launch the app.

3. Installing the app on your phone

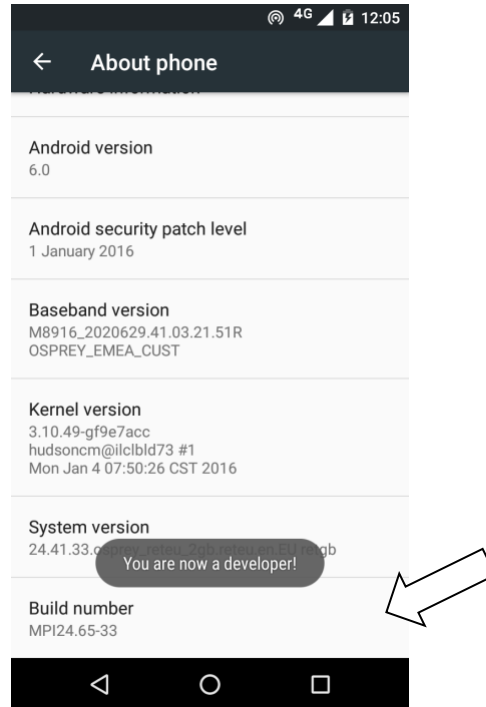
In the above section, you have seen how to compile an Android app. The result is an APK file, the installation file for an Android app. You now need to copy this APK file to your phone, install it and launch the app.

Here is how to do this:

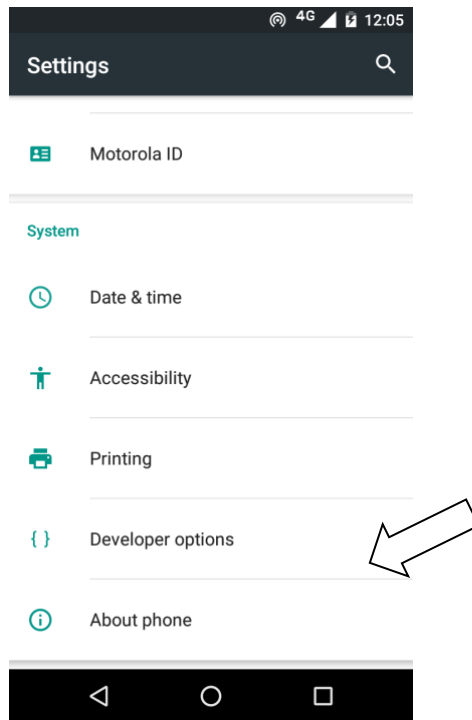
1. Connect your Android phone to your computer using a **USB data cable**.

(Sometimes you get cheap USB cables that can only charge a phone but cannot transfer data, so make sure you have the right kind of cable.)
2. Ensure that **Developer Options** ➤ **USB Debugging** is enabled on your phone. By default, on new phones, Developer Options is turned off. This is how you can enable it:
 - i. Open the **Settings** menu of your phone.
 - ii. Scroll down to the bottom of the menu and tap on **About Phone**.
 - iii. Find the **Build Number**. This could be on the About Phone page, or under a sub-menu such as 'Software Information'.

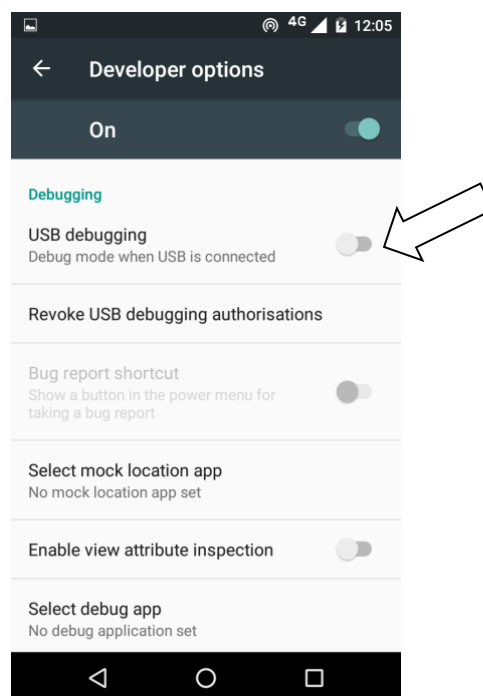
- iv. Tap on the Build Number **seven times**. As you do this, you will see a series of messages appearing: “You are now 3 steps away from being a developer”, “You are now 2 steps away from being a developer”, “You are now 1 step away from being a developer”, “You are now a developer!”.



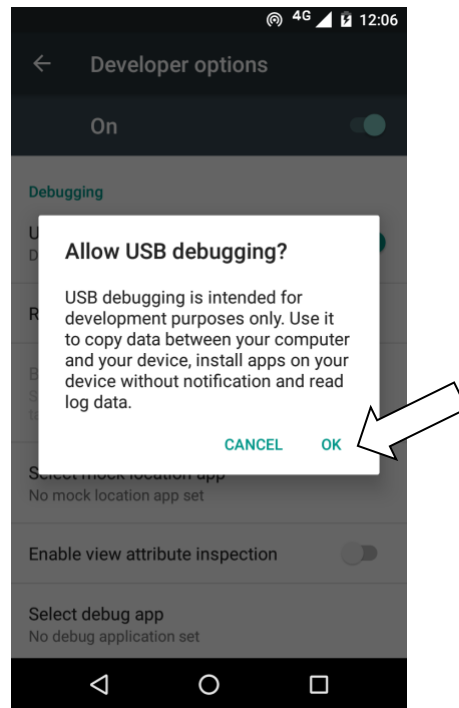
- v. Now return to the Configuration menu of your phone. Look for the Developer Options menu item. You might see **Developer Options** above the **About Phone** menu item. If you do not see it here, it could be in **System** settings, under **Advanced**. Different phones place Developer Options in different places, so look around your Configuration menu until you find it.



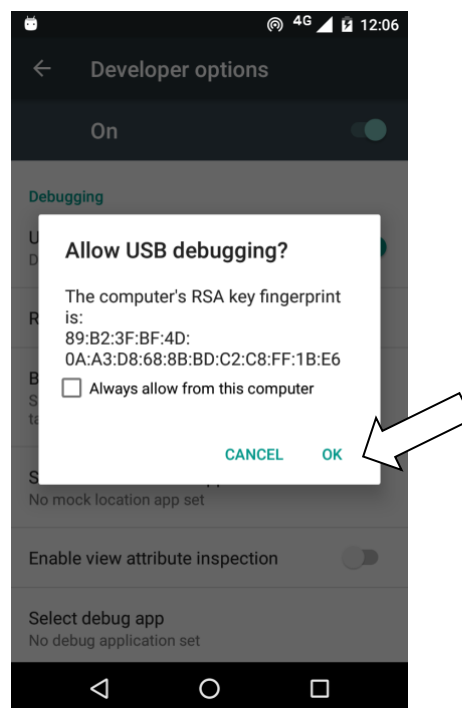
- vi. Tap on **Developer Options** and ensure that it is turned on.
- vii. Scroll down the Developer Options page and find **USB Debugging**. Enable this setting.



When you do this, you will probably get a message “Allow USB Debugging?”. Tap **OK**.



If you see a message box like this, tap **OK**:



3. In Scripture App Builder, click the **Install APK** button on the toolbar at the top right of the screen.

A window will open and the APK file will be copied to your phone and installed. The app will then be launched.

If this does not work, look at the command window to see if there is an error message. If you see a message such as “No devices/emulators found”, it means that your phone and computer are not connected correctly or that you have not enabled USB debugging on your phone.

Note:

Described above is a two-step process: **Build App** and then **Install APK**. If you prefer, you can tell Scripture App Builder to do this in one step, i.e. for the APK to be installed and launched automatically after building an app. See **Tools** ➤ **Settings...** ➤ **After Build** to enable this feature.

4. App Creation Basics

4.1. How should I choose the app package name?

The standard for an app package name is to begin with the reversed web address of the publishing organisation, e.g. if it is Wycliffe Bible Translators, the package name could begin with:

[org.wycliffe](#)

and will be followed by something identifying the language and type of publication (New Testament, Bible, etc.), e.g.

[org.wycliffe.cccc.nnnn.bible](#)

where ‘cccc’ is the country name and ‘nnnn’ is the language name.

If you work for a Bible translation or publishing organisation, you might have standards to follow for package names, so please contact your digital publications coordinator for advice on this.

Once you publish your app on an app store, you cannot change its package name later if you want users to continue to receive updates. The package name uniquely identifies the app. Those who install the app will be able to find its package name on their device. It will also appear in the web address for your app if you make it available on Google Play.

If you are building apps for **test purposes** on your devices, you can use a package name beginning with **com.example**, e.g.

[com.example.test.app123](#)

But remember to change it before you publish the app.

4.2. Where do the default book names and abbreviations come from?

If your books are in a Paratext project, the first place SAB will look for the book names is in the BookNames.xml file in the same folder as the USFM files. This file contains the names specified in the **Scripture Reference Settings** (found at **Project > Scripture Reference Settings...** within Paratext):

Book	Abbreviation (\toc3)	Short Name (\toc2)	Long Name (\toc1)
GEN - Genesis	Gen	Genesis	Genesis
EXO - Exodus	Exodus	Exodus	Exodus
LEV - Leviticus	Lev	Leviticus	Leviticus
NUM - Numbers	Num	Numbers	Numbers
DEU - Deuteronomy	Deut	Deuteronomy	Deuteronomy
JOS - Joshua	Josh	Joshua	Joshua
JDG - Judges	Judg	Judges	Judges
RUT - Ruth	Ruth	Ruth	Ruth
1SA - 1 Samuel	1 Sam	1 Samuel	1 Samuel
2SA - 2 Samuel	2 Sam	2 Samuel	2 Samuel
1KI - 1 Kings	1 Kgs	1 Kings	1 Kings
2KI - 2 Kings	2 Kgs	2 Kings	2 Kings
1CH - 1 Chronicles	1 Chr	1 Chronicles	1 Chronicles

If your books are not in a Paratext project or if the Scripture Reference Settings are blank, SAB will look for the book names within the book files themselves. The USFM standard is to use **\toc2** for the short book name and **\toc3** for abbreviations:

e.g.

\toc1 The Gospel according to Matthew	(long book name)
\toc2 Matthew	(short book name)
\toc3 Mat	(abbreviation)

If Scripture App Builder does not find **\toc2** or **\toc3**, it tries to guess the book name and abbreviation from **\mt** (main title) and **\h** (heading), but it is best to provide them in the toc fields.

For the Digital Bible Library text release bundles, the book names come from the metadata.xml file within the bundle.

You can change the book names and abbreviations on the book page for each book.

4.3. Do I have to create a new keystore for each app, or can I reuse the same keystore for several of my apps?

You can use the same keystore and key alias for all or several of your apps.

See here for more details:

<http://developer.android.com/tools/publishing/app-signing.html>

4.4. How can I include pictures in my apps?

You can include JPG or PNG image files in the text of the app. There are a few ways of doing this:

Method 1: Specify the image filenames within the Paratext USFM or USX files

The app recognises the USFM illustrations markers \fig.... \fig*:

\fig DESC|**FILE**|SIZE|LOC|COPY|**CAP**|**REF****\fig***

i.e. 7 parameters, separated by six vertical line characters.

The only required parameter is FILE – the filename. If the file extension is .tif, the app will look for a file with the same name but with a .png or .jpg extension instead.

You can optionally include captions (CAP) and chapter/verse references (REF).

Examples:

\fig |picture1.jpg| || |This is a picture|5:3\fig*

\fig |picture2.tif| || |This is another picture|\fig*

\fig |picture3.jpg| || ||\fig*

\fig picture3.jpg\fig*

Add the image files to the **Media ➤ Illustrations** page of Scripture App Builder so they can be included in the app.

Method 2: Include the images inside a Word document

If your book is specified by a Word document, you can embed the images directly in the document between the lines of text.

Method 3: Use the image placement feature

You can tell Scripture App Builder to place an image at a certain point in a text without modifying the source file for the text. To do this:

1. Go to the **Media ➤ Illustrations** tab.
2. Click Add Image and select the image file(s) to add.

3. Double-click on each image in the table, select the **Placement** tab, and click **Add Placement...** to specify where the image should be placed in the text.

The images will be included in the app assets and compiled into the final APK, so try to keep them as small as possible.

4.5. How can I add a horizontal line to separate certain paragraphs in text?

Use the marker \zhr to add a horizontal line. This will add a <hr> element in the HTML.

Please note that this is non-standard USFM, so this marker should not be used if you are submitting this same file to the Digital Bible Library.

4.6. I don't like the name "Scripture App". Have you thought of calling the app something else?

The program that allows you to define and build apps is called "Scripture App Builder" but the app itself doesn't have a name. It is up to you to choose the names for the apps you build.

You will not see "Scripture App" anywhere in the apps you create, so feel free to use an appropriate name in an international, national or local language. App names that work well in certain contexts will be less helpful in other contexts, which is why SAB allows you to contextualise it.

4.7. Can I build apps when I do not have internet access?

The first time you build an app, you will need to be connected to the internet otherwise the compiler will fail. It will download a set of libraries used by the Gradle compiler. After that you can set the 'offline' version in **Settings** so you can work offline.

If you want to be able to build your first app without needing internet access, it is possible to copy the Gradle cache files from another computer that has already downloaded them. This will only work, however, if the absolute path to the files is exactly the same on the computer from which the files are taken as on your computer, e.g. "C:\gradle" on computer A and "C:\gradle" on computer B. It will not work if you have "C:\Users\John\gradle" on computer A and "C:\Users\Jenny\gradle" on computer B (which is the default Gradle cache folder).

So, on computer A, to get the cache files to distribute:

1. Go to **Tools > Settings > Build Settings > Gradle Cache Folder**.
2. Enter "C:\gradle" and OK.
3. Build an app. The Gradle cache files will be downloaded to C:\gradle.

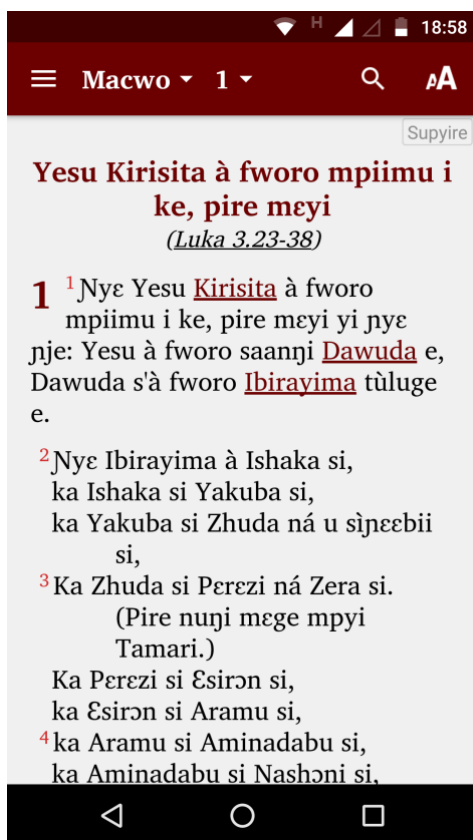
Then, on computer B:

4. Copy the C:\gradle folder from computer A to C:\gradle.

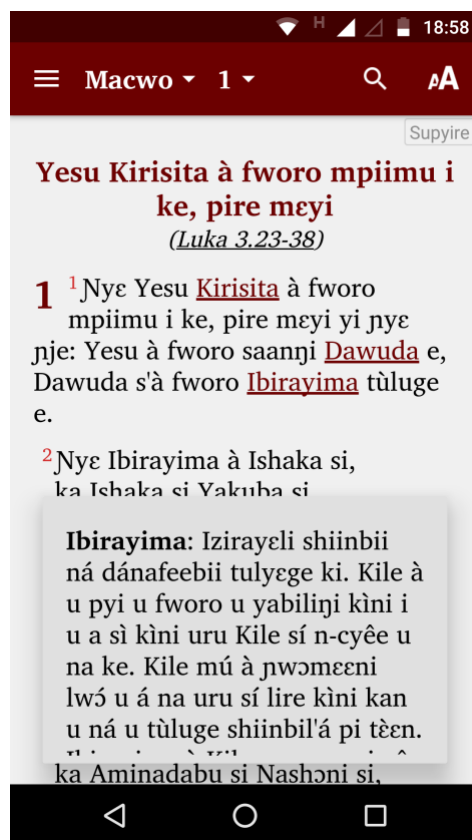
5. Go to **Tools** ➤ **Settings** ➤ **Build Settings** ➤ **Gradle Cache Folder**.
6. Enter "C:\gradle" and OK.

4.8. How do the glossary entry popups work?

It is possible to get the app to link words in the main Scripture text with glossary entries.



Glossary links in Scripture text



Glossary popup, shown after tapping "Ibirayima"

To get glossary links working, do the following:

1. Add a **glossary file** to the app as one of the books. A glossary file should have an id of GLO and each keyword should be marked with \k \k*

\id GLO
 \toc3 KkK
 \toc2 Kafilakaya Ya Kóri
 \c 1
 \p \k Aba\k*: Yawutuu shi jomɔ pu wo kafila kaŋa ka ki ɲɛ ke, ki kóri wu ɲɛ: «To», kelee «baa», kelee «baba». Pɛɛɲɛ mɛɛɛ ki ɲɛ kii, taanra ɲɛ kemu ni ge, lee na Yesu ya Kile yiri na «Aba» (Galasi Shɛɛn 4:6).
 \p \k Abeli\k*: Abeli bi bye Kayɛn wo cuun. Wee w'a pye Adama ni Awa yɛ wo ja shuun wo we. Kile ɲii fyaara ni wu bi ɲaari (Macoo 23:35; Luka 11:51; Heburuu 11:4).
 \p \k Ibirayima\k*: Kile ya shi wemu shɔɔnri yaha wuyɛ mu ge, wee w'a pye Yawutuu pu wo shi we. Ibirayima w'a pye pu sefelee pu beeri wo to n'a daa wu ni.

2. Mark words in the **main text** with \w \w*

\c 1
 \s Yesu Kirisa nɔhɔ tayirige sɛmɛ
 \r (Luka 3:23-38)
 \p
 \v 1 Yesu Kirisa nɔhɔ tayirige sɛmɛ w'a we. Yesu ya foro saannaa
 \w Dawuda\w* shi ni, Dawuda d'a foro \w Ibirayima\w* shi ni.
 \li1
 \v 2 Ibirayima w'a \w Ishaaga\w* se.
 \li2 Ishaaga di \w Yakuba\w* se.
 \li2 Yakuba di Zhuda ni wu ceboronamaa se.

If the glossary form is different from the form of the word in the text, use the vertical bar | character to specify the glossary form.

e.g. where 'angel' is in the glossary file:

\v 5 You have made them a little lower than the \w angels|angel\w* and crowned them with glory and honour.

4.9. Can I build an app from the command line?

Yes, Scripture App Builder has a command line interface which allows you to create a new app and build it, or load an existing app and build it.

The command line tool is named **sab** and can be found in the Program Files folder, usually c:\Program Files (x86)\SIL\Scripture App Builder.

sab takes the following parameters:

Option	Description
-new	Create a new app project

-load <project>	Load an existing app project
-build	Build app project (use with either -new or -load)
-no-save	Do not save changes to app (use with -load)
-?	Show command line help
-n <app-name>	Set app name. Enclose the name in "double quotes" if it contains spaces.
-p <package-name>	Set package name, e.g. com.myorg.language.appname
-b <filename>	Add book or bundle file. This could be a USFM file or a zipped set of USFM files. It could also be a Digital Bible Library text release bundle.
-i <filename>	Include additional parameters file. Use the full path of the file and enclose it in "double quotes" if there is a space in the path.
-a <filename>	Set about box text, contained in text file. Use the full path of the file and enclose it in "double quotes" if there is a space in the path.
-f <fontname>	Set font name or filename, e.g. "Charis SIL Compact", "c:\fonts\myfont.ttf" The font name must be one of the items in the list of fonts in the New App wizard. For other fonts, specify the full path to the font filename.
-g	Use Grandroid
-ic <filename>	Add launcher icon (one or more .png files). Use the full path of the files and enclose them in "double quotes" if there is a space in the path.
-l <lang-code>	Set language for menu items and settings, e.g. en, fr, es
-ft <feature=value>	Set a feature, e.g. book-select=grid
-vc <integer>	Set version code, e.g. 1, 2, 3, or +1 to increment the current version code by 1.
-vn <string>	Set version name, e.g. 1.0, 2.1.4, or use +1, +0.1, +0.0.1 to increment the current value.
-ks <filename>	Set keystore filename. Use the full path of the file and enclose it in "double quotes" if there is a space in the path.
-ksp <password>	Set keystore password
-ka <alias>	Set key alias

-kap <password>	Set key alias password
-fp <folder=path>	Set a folder path, e.g. "app.builder=c:\Scripture App Builder".

Examples:

```
sab -new -n \"My App\" -p com.example.myapp -b MyBookBundle.zip
      -f \"Charis SIL Compact\" -i keys.txt -build
```

```
sab -load \"My App\" -build
```

5. Fonts

If you are using a non-Roman script or a Roman script with combining diacritics, some Android devices will not display your fonts correctly. To overcome these problems, try using the Grandroid and/or GeckoView libraries.

Library	Purpose	Android Versions Supported	Additional Size
Grandroid	Rendering complex fonts and combining diacritics correctly in older versions of Android	Android 4.0, 4.1, 4.2, 4.3	200 KB
GeckoView	Rendering complex fonts correctly in most versions of Android, especially fonts that are Graphite-enabled	Android 4.1 and above	55 MB

You can configure these on the **Appearance > Fonts > Font Handling** page.

5.1. What is Grandroid?

Grandroid (Graphite for Android) is a collection of native libraries from SIL Writing Systems Technology (WSTech). They can be packaged within the app, enabling older Android devices (versions 4.0 to 4.3) to make use of Graphite font rendering features.



Grandroid is not only about Graphite. It also fixes a few font display problems.

You can find more information about Graphite here:

http://scripts.sil.org/cms/scripts/page.php?site_id=projects&item_id=graphite_home

You can find more information about Grandroid here:

<https://github.com/silnrsi/grandroid>

5.2. When do I need to include the Grandroid libraries?

This will depend on the font and special characters you need to display. The more complex your script, the more likely you are to need Grandroid support.

Please note that if a font displays correctly on your own phone without Grandroid, it does not mean it will display correctly on all phones and Android versions. As well as testing your app on the latest version of Android, it would be a good idea to test it on a phone running Android 4.2 or 4.3 (which have known font display problems).

You will almost certainly need to include the Grandroid libraries:

- If you have a non-Roman script, e.g. Greek, Cyrillic, Armenian, Hebrew, Arabic, Syriac, Thaana, Devanagari, Grumukhi, Oriya, Tamil, Telugu, Kannada, Malayalam, Sinhala, Thai, Lao, Tibetan, Myanmar, Georgian, Hangul, Ogham, Runic, Khmer, Ethiopic or Nko.
- If you have a Roman script which makes use of combining diacritics, such as separate acute accents or tone marks (e.g. $\acute{\text{u}}$, which is composed of two characters, but not $\acute{\text{e}}$ which is a single character).

You are unlikely to need to use Grandroid:

- If you have a simple Roman script which does not make use of combining diacritics. So that means a-z, plus other IPA characters such as ϵ , ɔ , ɲ , η , etc. as long as they are not being combined with tone marks or accents.

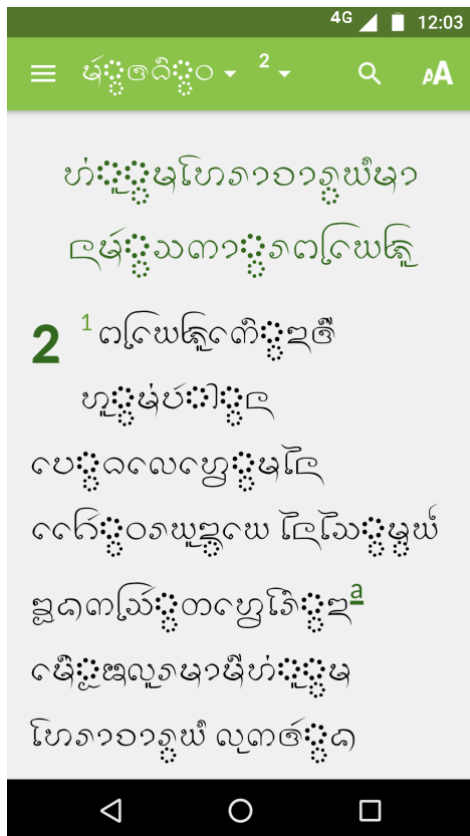
If you try and display a complex script without Grandroid, you might find the following problems:

- The system font being used rather than the font you specify - on Android 4.2 and 4.3 (Jelly Bean).
- Lines with combining diacritics being displayed in the system font, while other lines are being displayed correctly - on Android 4.2 and 4.3 (Jelly Bean).
- A blank screen where there should be text - on Android 4.2 and 4.3 (Jelly Bean).

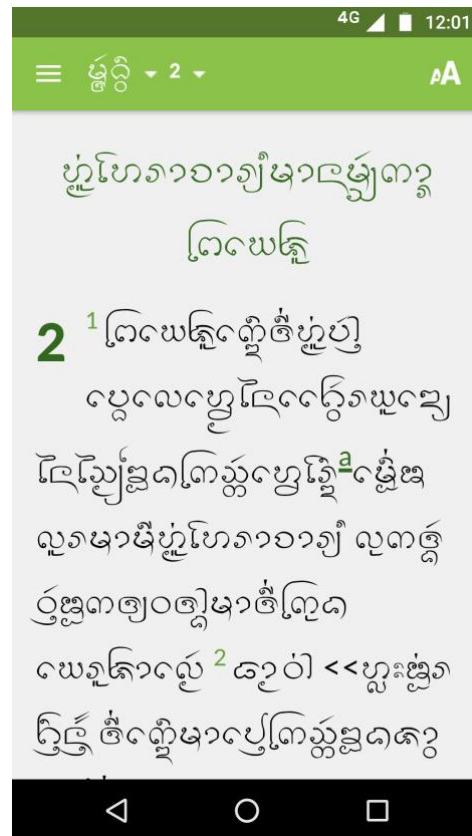
5.3. What is GeckoView?

GeckoView is a viewer component from Mozilla that replaces the standard Android viewer for Android versions 4.1 and above. It can render Graphite fonts correctly.

In the screenshots below, the standard viewer is used in the left image and GeckoView is used in the right image. Notice the dotted circles on the left, indicating that the font is not being rendered properly.



Using the standard viewer



Using GeckoView

The required GeckoView library files will add at least 55 MB to your app size, so do not enable GeckoView unless you know you need it to display your fonts correctly.

You will find that when you build an Android APK with GeckoView, the APK size will be at least 200 MB larger than without GeckoView. This is because it contains the GeckoView libraries for four different device architectures (32-bit ARM, 64-bit ARM, 32-bit Intel and 64-bit Intel). In practice, your app users do not need to install such a large file.

- For **online app distribution**, you need to upload an AAB file (app bundle) to Google Play rather than an APK. The AAB file contains all the GeckoView libraries for all four device architectures, but when a user installs an app from Google Play, Google will create a tailored APK for them, with just the libraries and components their phone needs. The AAB file might be over 300 MB, but the actual size of the app for any user will be much smaller.
- For **offline app distribution**, you can ask for Scripture App Builder to create multiple APKs, one for each device architecture. Do this on the **App** > **APK** tab. Each of these APKs will be significantly smaller (around 55 MB extra for GeckoView). You just need to choose the right one(s) to distribute, according to the types of phones people have. Many of the phones today will use the 64-bit ARM APK. Otherwise, the 32-bit ARM APK is used for most older phones. Intel phones are less common, but it will depend on the phones being sold in your country.

6. Audio

6.1. How can I associate audio files with the text?

Scripture App Builder allows you to use two different types of audio file:

- **Audio files that correspond to a page of text in the app**, e.g. the audio recording of a chapter of the Bible or the text in a picture story book. These can be linked to timing files to enable synchronized highlighting of the text when the audio is playing.
- **Short audio clips** which are played when the user taps a linked word, phrase or image in the app.

The first of these types of file is the most common in SAB apps and the instructions in this section will refer to these. For more information on using audio clips, please see section 6.5.

To associate audio with pages of text in your app, you will need one audio file per chapter. For example, if you have a book 'Matthew' with 28 chapters, you will need 28 audio files, one for each chapter.

Matthew chapter 1	Bible-MAT-01.mp3
Matthew chapter 2	Bible-MAT-02.mp3
Matthew chapter 3	Bible-MAT-03.mp3, etc.

If you have several audio files per chapter or several chapters in one audio file, you will need to concatenate or split your files to create files of one chapter each before SAB can use them. This could be done using [Audacity](#).

For picture story books, you can choose whether you have one audio file per page, or a single audio file for the whole book.

To add audio files:

1. Go to the **Audio Files** tab for the book, or the **Audio Files** tab for the book collection.
2. Click the **Add Audio Files...** button.
3. Select the audio files to add.

SAB will try to match the audio files to the chapters in the book(s), so you should use audio filenames that will make it obvious which file corresponds to which book and chapter.

The optional timing files can be added on the same page: one timing file per audio file.

6.2. How do I create the audio timing files for audio-text synchronization?

The timing files tell the app when to begin highlighting each verse or phrase.

There are three ways of creating these files:

1. Create the timing files manually using the **Audacity** audio editing program. For instructions, please refer to the document *Using Audacity for Audio-Text Synchronization*.
2. Automate the creation of the timing files using **aeneas**. For instructions, please refer to the document *Using aeneas for Audio-Text Synchronization*.
3. Use **HearThis** to record the audio. When you export the MP3 files, there is an option to export a set of timing files at the same time. For instructions, please refer to the document *Using HearThis for Audio Recording*.

6.3. How do I distribute the audio files with the app?

There are four ways of including audio files in your app: assets, external folder, internet download or FCBH download. You can use a single **file source** for all of the files in an app or you can combine two or more sources in an app.

To specify the file source(s) in Scripture App Builder, you need to visit the following two pages:

1. The **Media ➤ Audio ➤ File Source** page, which defines the available file sources.
This page can be found in the apps tree view just under **Analytics** on the top level of app pages. You can modify, add and remove file sources here.
2. The **Audio Files** tab for a book or book collection, which lists the audio files with their corresponding file source.
To change the source for a file or files, select the rows you want to change and select **Change Source**.

The follow sections describe the different file source types.

1. Assets

The audio files will be packaged inside the APK file for the app. This is the easiest method for a few files (e.g. one book) and requires no permissions. But be beware that the APK will get very large if you have several books of audio. The maximum size of an app that can be installed from the Google Play store is 150 MB.

2. External Folder

No audio files are packaged within the app, so the APK is small. The app will look in a specified folder to find the audio files it needs. If you are distributing the app via memory card, you include the folder of audio files on the memory card together with the APK. This method requires the 'Read external storage' permission but not internet access.

You can place the audio files inside sub-folders and sub-sub-folders in the specified folder, using any folder names you choose.

For example, here are some possible folder names:

```
\Bible Files
  \1 Old Testament
    \01 Genesis
    \02 Exodus
  \2 New Testament
    \01 Matthew
    \02 Mark
```

or

```
\NT Mamara
  \01 Macoo
  \02 Marika
  \03 Luka
  \04 Yohana
```

There is no need to use 01, 02, etc. in the folder names, but these can be useful if folders are listed in alphabetical order

Alternatively, you can place all the audio files in a single folder without using any sub folders.

If the app does not find audio files in the specified folder or its sub-folders, it will also search the other folders on the device to see if it can find them there. For example, if the specified folder name is 'Audio 123' but the files are located in the 'Audio 456' folder instead, the app should find them. Once it has found a folder with a needed audio file, it will keep a note of it so it knows where to look next time.

3. Internet Download

Like method 2, no audio files are packaged within the app, so the APK is small. The app will look in a specified SD card folder to find the audio files it needs. If it doesn't find them there, it will look in all the other folders on the device. If it still cannot find them, the app can download the files one by one when it needs them from a website of your choice. This method requires the 'Read external storage', 'Write external storage', 'Connection state' and 'Internet' permissions.

Audio filenames

The internet download works best if your audio filenames do not include any spaces. A filename of the form "LNG-MAT-01.mp3" is better than "LNG MAT 01.mp3".

Audio file hosting

Recommended storage locations for the files on the internet include:

- **A language-specific website**

If you have a language-specific website for making resources available for download, you could place the audio files in a folder on the website.

For example, if your website is called 'www.ourlanguage.org', you could upload the audio files to a folder called 'audio'. The http address for your audio files would then be: <http://www.ourlanguage.org/audio>

Your website administrator should be able to help you do this.

- **Cloud Storage Services**

Amazon S3 (Simple Storage Service): <http://aws.amazon.com/s3/>

Backblaze B2 Cloud Storage: <https://www.backblaze.com/b2/cloud-storage.html>

Google Cloud Storage: <https://cloud.google.com/storage/>

These cloud storage services are designed for fast, reliable and secure online storage. Once you have created an account, you create a 'bucket' in which to place your audio files. When you add the files, you need to make them public and make a note of the web address link to use to access them, e.g. <http://s3.amazonaws.com/yourbucketname>

You will get some months of free storage before there is a charge according to the bandwidth used, i.e. how many MB of audio users download. It might be easiest to organise this kind of cloud storage at an organisational level rather than creating a new account for each language.

4. FCBH Bible Brain

If the audio files that your app uses come from Faith Comes By Hearing (FCBH), there is no need to upload them to the internet. They are already online and can be accessed via [Bible Brain](#). This is the server used by the [Bible.is](#) website and the Bible.is mobile

app. Take a look at the Bible.is website to see if the language you are interested in is already available.

If you have any non-FCBH Scripture recordings, those you have recorded in-house with the help of your own organisation's technicians, FCBH is willing to consider hosting these files as well in Bible Brain. This applies, for example, to good quality recordings of complete Bible books – but not Bible story portions or teaching materials. To investigate this option, please contact your area FCBH representative.

For your app to be able to download files from Bible Brain, you will need to register with FCBH and provide the app with your Bible Brain **API key**, received on registration, and the **Fileset ID** of the audio files. This method requires the 'Read external storage', 'Write external storage', 'Connection state' and 'Internet' permissions.

When building your app, you will need to have downloaded the FCBH audio files onto your computer and add them to the Audio Files page. This is so SAB can see the filenames and allow you to create corresponding timing files for text-audio synchronization. The files are used for the app building process and will not be packaged within the APK file.

Getting an API Key

The Bible Brain API Key is a 32-character alpha-numeric string.

To get a key, you need to register with FCBH:

1. Go to <https://biblebrain.com>
2. Click **Get Your API Key**.
3. On the **Request Your API Key** form, fill in your details, mentioning that you are building apps with *Scripture App Builder* and click **Submit**.

Getting the Fileset ID

After you get your API key from FCBH, you will need to look for the **Fileset ID** for the audio files you need.

The Fileset ID is a 10-character string which uniquely identifies the set of audio files for a language, e.g. "MYKWBTN2DA" (for the Minyanka NT), "TPIPNGN2DA" (for the Tok Pisin NT).

To find the Fileset ID for a language and Bible translation:

1. Go to the **FCBH MP3 Downloads** page:
<https://www.faithcomesbyhearing.com/audio-bible-resources/mp3-downloads>
2. Enter the **Language** and select a **Version**.
3. Look at the URL in your web browser's address bar. The last 10 characters are the Fileset ID.

Here is an alternative method to find the Fileset ID:

1. Go to the **Bible Brain** page of the FCBH website using <https://biblebrain.com> and click on the **Developer Docs** button.
2. Go to the **Available Content** page. You can find this in the menu on the left of the screen.
3. Enter the language or country name to filter the search results.
4. Click on the language name in red to go to the language page on the Bible.is website.
5. Take a look at the URL in the address bar to find the **first six characters** of the Fileset ID. This will include the language code and version, e.g.

MYKWBT – Minyanka (Wycliffe Bible Translators)

BBOBSM – Konabere (Bible Society of Mali)

6. The **seventh** character of the Fileset ID is the **collection type**:¹
 - **C** - Complete (all books and chapters of both testaments)
 - **N** - New Testament (all books and chapters)
 - **O** - Old Testament (all books and chapters)
 - **P** - Partial (incomplete set of books or chapters)
 - **S** - Story (Bible-related content, not a formal translation, not usually organized by book or chapter)
7. The **eighth** character of the Fileset ID is the **drama type**:
 - **1** - “non-drama”, usually multi-voice without background music, but sometimes single-voice
 - **2** - “drama”, meaning that it is multi-voice with background music.
8. The **ninth and tenth** characters are the **media type**, which will be “**DA**” for digital audio.

The combined Fileset ID will have 10 characters, e.g.:

- MYKWBTN2DA – Minyanka, Wycliffe, New Testament, Drama, Audio
- BBOBSMN1DA – Konabere, Bible Society of Mali, NT, Non-Drama, Audio

Configuring your app to use the API Key and Fileset ID

Once you have an **API Key** and a **Fileset ID**:

1. Go to **Media ➤ Audio ➤ File Source**.
2. Click **Add File Source**.

¹ <https://www.faithcomesbyhearing.com/bible-brain/core-concepts>

3. Select “FCBH Bible Brain” as the **Source**.
4. Enter your API Key and Fileset ID in the fields provided.

Edit File Source

Specify the details of this file source.

Name/Description: ☒ Default

Source:

Specify how the app will access the online media files.

Access Method:

Specify the parameters for the FCBH Bible Brain. If you do not have a key, [Request](#) an API Key. If you test your key and it does not work, then click Contact Support.

API Key:

Fileset ID:

Download to Folder:

5. Click the **Test** button to verify that the API Key and Fileset ID are valid and that they will enable your app to access the required audio files.
6. You can modify the **Access Method** to specify whether app users will be able to download and/or stream the audio files. You might find that only streaming is allowed for certain Fileset IDs, due to licensing restrictions.

6.4. How are the timing files distributed for the app?

If you have timing files for audio-text synchronization, they are always packaged in the app assets whatever the audio source setting chosen in 6.3 above. They are not downloaded from anywhere.

If timing files are not available when the app is compiled and are subsequently created by someone who isn't building the app (such as a volunteer), they can be placed in the external folder for testing. For more details, please see section 6.7 in the document *Using Audacity for Audio-Text Synchronization*, which answers the question “I’d like to give someone else the task of creating timing files. Is there any way of them testing these without needing to know how to build the app?”

6.5. How can I use audio clips in the app?

Audio clips are short audio files which are played when the user taps a linked word, phrase or image in the app.

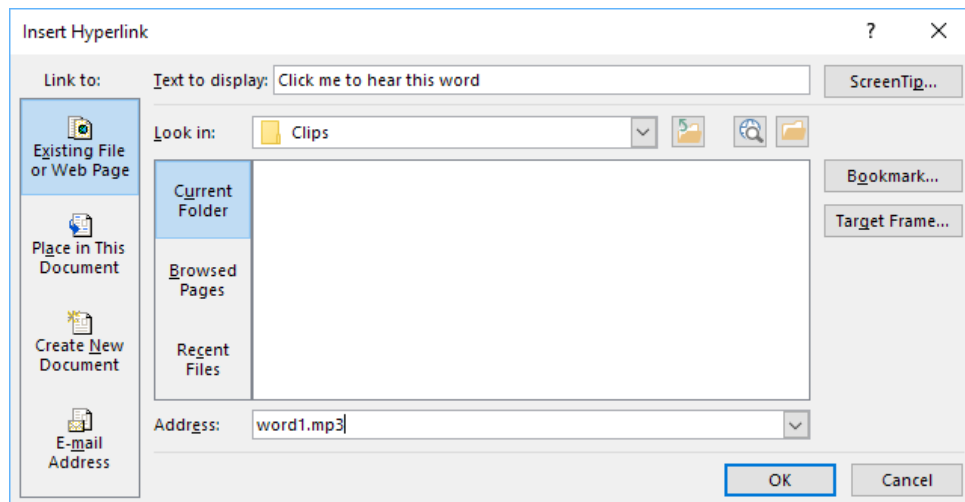
To include an audio clip in your app:

1. Select the **Audio** page for the app, and go to the **Audio Clips** tab.
2. Click **Add Audio Clips...** and select the audio file you want to add.
3. Add a hyperlink to this audio file in your text.

You can do this using the markdown format with [] and (), e.g.

[Click me to hear this word](word1.mp3)

or if your text is in a Word document, select a word or a phrase, right-click, **Hyperlink...** and enter the audio filename as the address to link to.



Specify the audio filename without a path, e.g. “word1.mp3” rather than “C:\My clips\word1.mp3”.

6.6. How can I create subtitles from text and timing files?

If you have text and timing files for audio synchronisation, Scripture App Builder can create subtitle files for you. These can be of one of the following four formats:

- SubRip SRT subtitles (.srt)
- SAMI subtitles for Windows Media Player (.smi)
- WEBVTT Web Video Text Tracks (.vtt)
- Adobe Encore Text Scripts (.txt)

To create a subtitles file, go to the **Audio Files** tab, right-click and select **Export Subtitles**.

7. Video

7.1. How do I include videos in the app?

The app can handle two types of video:

- **Videos hosted online for streaming**, e.g. YouTube, Vimeo, Jesus Film videos, and MP4 streaming video files.
- **Videos for offline use**, MP4 files which can be packaged within the app assets, distributed on memory cards, or downloaded from the internet for use and sharing offline.

To add an **online** video:

1. Select the **Video** page in the app builder.
2. Click **Add Videos...**
3. Select **Online streaming video, from YouTube, Vimeo, etc.**, and click **Next**.
4. Enter the video URL in the Online Address field, e.g.
<http://www.youtube.com/watch?v=ABcDEdGHIJ>.
5. Click **Finish**.
6. Double click on the video item in the table.
7. Specify a title (optional).
8. Select or modify the thumbnail image on the **Thumbnail** tab.

To add an **offline** MP4 video file:

1. Select the **Video** page in the app builder.
2. Click **Add Videos...**
3. Select **Video files in MP4 format**, and click **Next**.
4. Click **Add Videos...** and select the MP4 video files to add.
5. Click **Finish**.
6. Double-click on each video to edit its title (optional) and thumbnail image.

The next thing you need to do is tell the app builder where to insert and display the videos.

7.2. How do I reduce the size of a video that I want to package within the app?

If you want to include MP4 video files within the app, you need to make sure that they are as small as possible, otherwise your app will become too big to publish.

There are some good video compression tools online which can reduce the size of a video considerably without losing much quality. An example is the Clideo Compress Video tool, at <https://clideo.com>.

7.3. How do I specify where the videos will be displayed within the app?

There are three ways of inserting videos into the text:

- **Placement within Scripture:** Use the third tab, **Placement**, in the Edit Video dialog to specify a position within a chapter of Scripture: before or after a given verse or at the top or bottom of a page.
- **Video markers in SFM book files:** Use the `\video` marker, together with the ID of the video within an SFM book file, e.g.

```
\s This is a sub heading
\p Here is an example video to view:
\video V1
\p Here is another video:
\video V2
```

- **Video markers in Word documents:** Use the word **VIDEO:**, together with the ID of the video within a book's Word document, e.g.

```
This is a sub heading
Here is an example video to view:
VIDEO: V1

Here is another video:
VIDEO: V2
```

The **VIDEO:** tag should be at the start of a new paragraph, and include a colon (:) before the video ID.

7.4. How can I create slideshow videos using audio and timing files?

Slideshow videos are videos created using a set of still images together with audio files from a *Scripture App Builder* project. While the audio plays, the images will pan and/or zoom. Subtitles will be generated from the text in the app.



To create slideshow videos, please see the documentation manual #12, "Creating Slideshow Videos".

8. About Page

Every app must have an About page, which is displayed to the user when they select **About** from the bottom of the app's navigation drawer menu.

To define the About page for your app:

1. Select **About** in the apps tree view on the left of the screen.
2. Enter the text to be displayed on the app's About page.
3. Select the **Viewer** tab to see how the text will appear in the app.

8.1. What information should I include in the About page?

The About page can contain any information. What you include is your choice, but commonly included items include:

- The app name and version
- The name of the organization publishing the app
- Copyright and license information for the contents of the app (texts, images, audio). In most cases, the license agreements will require you to include such information.
- Link to your website
- Link to a privacy policy
- Contact information, such as an email address or phone number.

8.2. Which formatting codes can I use in the About page?

You can use the following formatting codes:

Bold text	Surround the text you want in bold with <code></code> and <code></code> markers. Example: <code></code> This is in bold <code></code>
Italic text	Surround the text you want in bold with <code><i></code> and <code></i></code> markers. Example: <code><i></code> This is in italics <code></i></code>
Underlined text	Surround the text you want in bold with <code><u></code> and <code></u></code> markers. Example: <code><u></code> This is underlined <code></u></code>
Website links	Use the format <code>[text](url)</code> , where 'text' is the text to display and 'url' is the web address. Example: Here is our <code>[website](http://www.example.com)</code>

Email links	<p>Use the format <code>[text](mailto:address)</code>, where 'text' is the text to display and 'mailto:address' contains the email address.</p> <p>Example: Contact us by <code>[email](mailto:contact@example.com)</code></p>
Phone numbers	<p>Use the format <code>[text](tel:number)</code>, where 'text' is the text to display and 'tel:number' contains the number to call.</p> <p>Example: Our number is <code>[012-345-678](tel:012345678)</code></p>
Image	<p>To add an image, first add the image file to the app illustrations (Images ➤ Illustrations) and use the following HTML code:</p> <pre></pre> <p>You can specify width information, as a percentage (e.g. "80%") or a fixed number of pixels (e.g. "80px"):</p> <pre></pre>
Left, right or centered alignment	<p>To align text or images, surround them with the following <code><div></code> markers:</p> <pre><div align="center">This is centered</div></pre> <pre><div align="left">This is left aligned</div></pre> <pre><div align="right">This is right aligned</div></pre> <p>To center an image:</p> <pre><div align="center"></div></pre>
Styles	<p>To apply styles (defined in the Styles page):</p> <pre><div class="q2">This is styled as poetry</div></pre>
Fonts	<p>To apply specific fonts to text, first make sure that the fonts are specified on the Fonts page. Then use the following syntax:</p> <pre>This is in font1</pre> <pre>This is in font2</pre>

8.3. Which variables can I use in the About page?

You can use certain variables on the About page. These are replaced by text in the app. For example, the variable `%version-name%` will be replaced by the app version number, e.g. "1.2" in the app.

App name	To display the app name, use the variable <code>%app-name%</code>										
App version	To display the app version, use the variable <code>%version-name%</code>										
Copyright and Licensing	<p>To display copyright and licensing information for text, audio and images:</p> <table> <tr> <td><code>%copyright-all%</code></td><td>all copyright information for all book collections</td></tr> <tr> <td><code>%copyright-text:CCC%</code></td><td>text copyright for collection CCC</td></tr> <tr> <td><code>%copyright-audio:CCC%</code></td><td>audio copyright for collection CCC</td></tr> <tr> <td><code>%copyright-images:CCC%</code></td><td>images copyright for collection CCC</td></tr> <tr> <td><code>%copyright-all:CCC%</code></td><td>text, audio and images copyright for collection CCC</td></tr> </table> <p>Replace CCC by the ID of the book collection, e.g. "C01". The information comes from the Copyright & Licensing tab for that book collection.</p>	<code>%copyright-all%</code>	all copyright information for all book collections	<code>%copyright-text:CCC%</code>	text copyright for collection CCC	<code>%copyright-audio:CCC%</code>	audio copyright for collection CCC	<code>%copyright-images:CCC%</code>	images copyright for collection CCC	<code>%copyright-all:CCC%</code>	text, audio and images copyright for collection CCC
<code>%copyright-all%</code>	all copyright information for all book collections										
<code>%copyright-text:CCC%</code>	text copyright for collection CCC										
<code>%copyright-audio:CCC%</code>	audio copyright for collection CCC										
<code>%copyright-images:CCC%</code>	images copyright for collection CCC										
<code>%copyright-all:CCC%</code>	text, audio and images copyright for collection CCC										
Program Type	To display the program type (SAB), use the variable <code>%program-type%</code>										
Program Version	To display the version of the program used to create the app, use the variable <code>%program-version%</code>										

9. Footers

It can be useful to display a footer block at the bottom of each chapter of text. This can contain information such as:

- The name of the Bible translation
- Copyright and licensing details
- Contact email addresses and phone numbers
- Website links for more resources or to order printed materials
- Link to discussion groups on social media

Footers are defined on the **Footer** tab for each book collection. You will find details of the formatting codes you can use, which are similar to those you can use on the About page (see the previous section).

10. Multiple Translations

You can include more than one Bible translation or script in your app, for example:

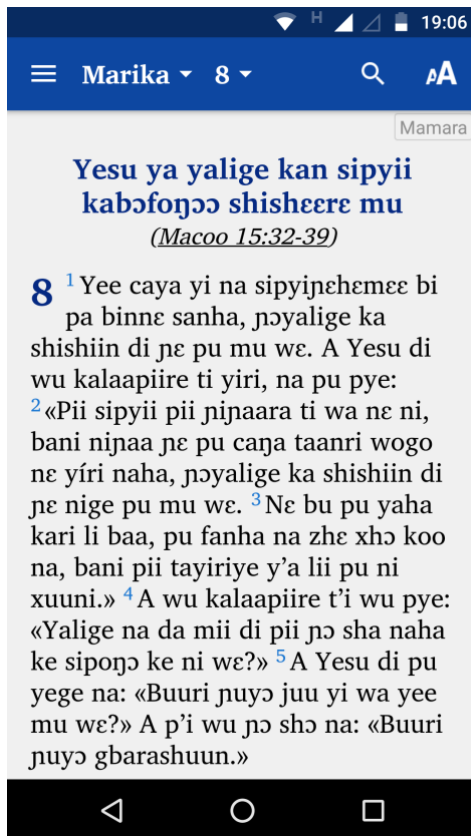
- A local language translation together with a translation for a neighbouring language.
- A local language translation together with a translation in a major language such as English or French.
- A translation in Roman script together with the same translation in a different script such as Arabic.
- A translation together with the original language texts in Hebrew or Greek.

You are not limited to two translations. You can choose to include three or more translations if you need to. However, the more translations you include, the bigger your app will become.

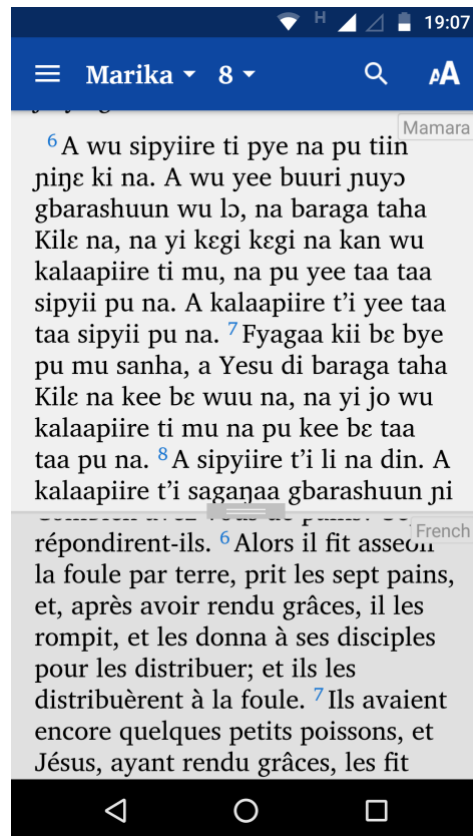
10.1. Three different layouts

Multiple translations can be displayed in the app in one of three layouts:

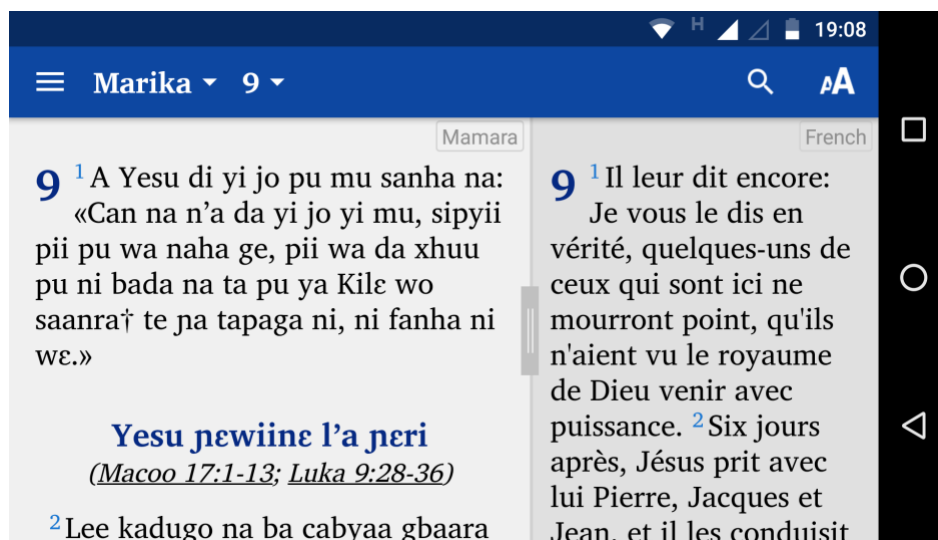
1. **Single Pane:** view one translation at a time, in the same way as you would in an app with a single translation.
2. **Two Pane:** view two translations at a time. In portrait mode, the translations will be displayed one on top of the other. In landscape mode, the translations will be displayed side by side. The user can drag the separator line up/down or left/right to change the amount of screen space allocated to each translation.
3. **Verse-by-Verse:** view two or three translations at a time, with each being displayed verse by verse, i.e. verse 1 in translation A, followed by verse 1 in translation B, verse 2 in translation A, verse 2 in translation B, etc. The different translations will be displayed in different colour fonts.



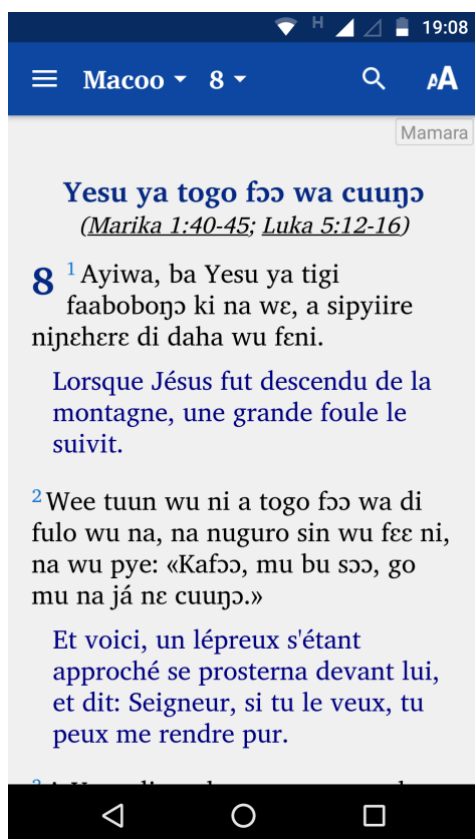
Single Pane Layout



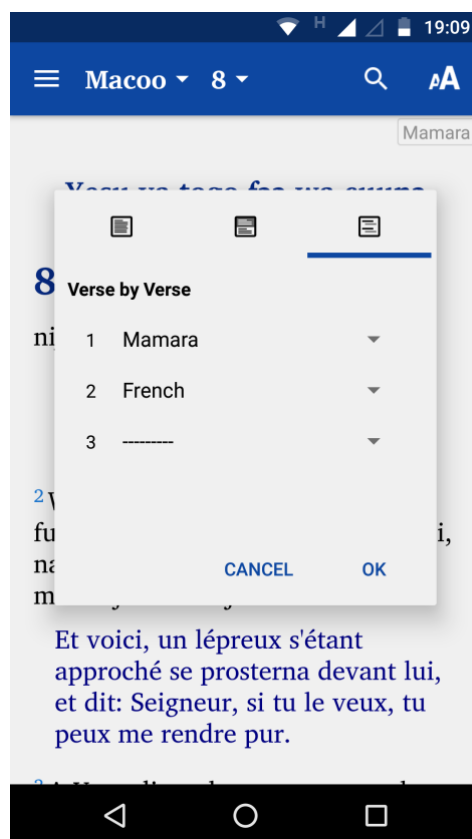
Two Pane Layout (portrait)



Two Pane Layout (landscape)



Verse-by-Verse Layout



Layout Configuration

Within the app, you can choose to give the user the permission to configure these layouts. To access the layout configuration dialog, the user taps on the name of the language at the top right of the viewer.

10.2. How to add a translation to the app

To add a translation to the app within Scripture App Builder:

1. Select the **Books** node for the app in the treeview on the left of the SAB screen.
2. Click **Add Book Collection...** to launch the Add Book Collection wizard.
3. On the **Book Collection Name** page, enter a name and short name for the new book collection.
Click **Next** to move to the next page.
4. On the **Books** page, click **Add Books** to add the book files to the book collection.
Click **Next** to move to the next page.
5. If you have selected Scripture books, the next page will be the **Book Order** page. Choose the book ordering according to the church tradition of the target users.
Click **Next** to move to the next page.

6. On the **Copyright and Licensing** page, specify the copyright and licensing information that you would like to appear on the About box in the app for this book collection. Use the **Copyright Helper** wizard to help you.

If you do not know what to put here, please ask the publishing department of your organisation for advice. They will want to make sure you get this right and do not simply make a guess as to what to include.

Click **Next** to move to the next page.

7. On the **Restrictions** page, specify if there are any restrictions on how this book collection can be used within the app.

Click **Finish** to close the wizard.

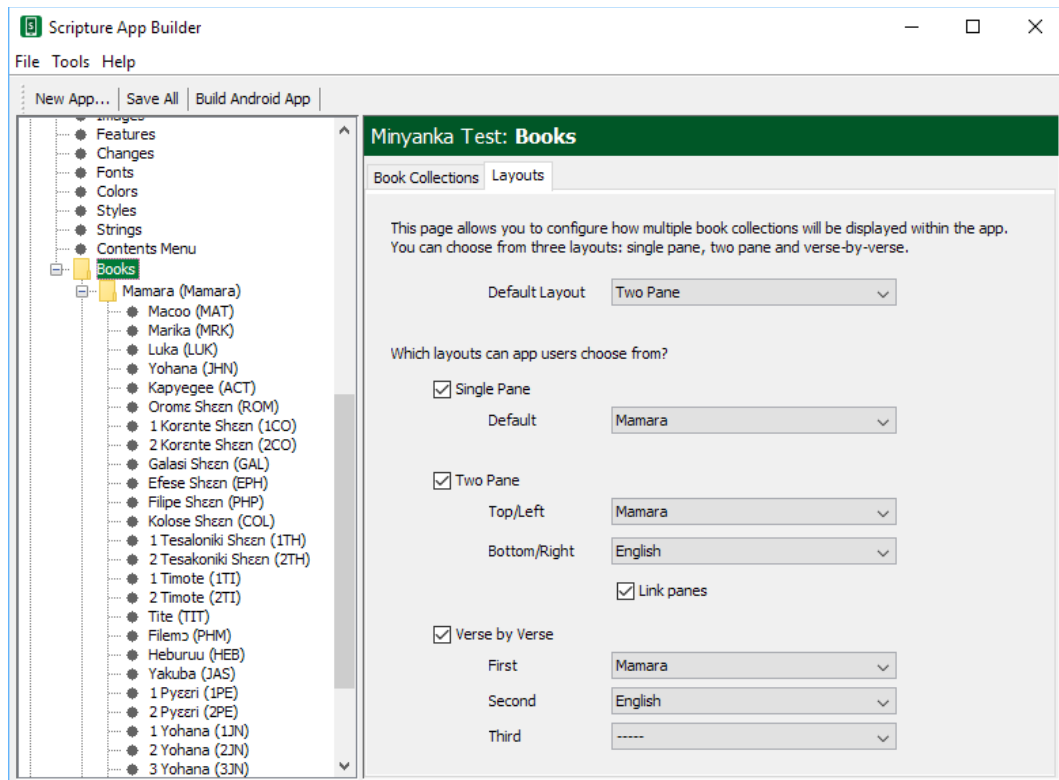
8. Configure the **Styles** for this new book collection on the book collection Styles page.

9. Go to the **About** box page and add the copyright and licensing information for this book collection.

10.3. How to configure the Layouts dialog

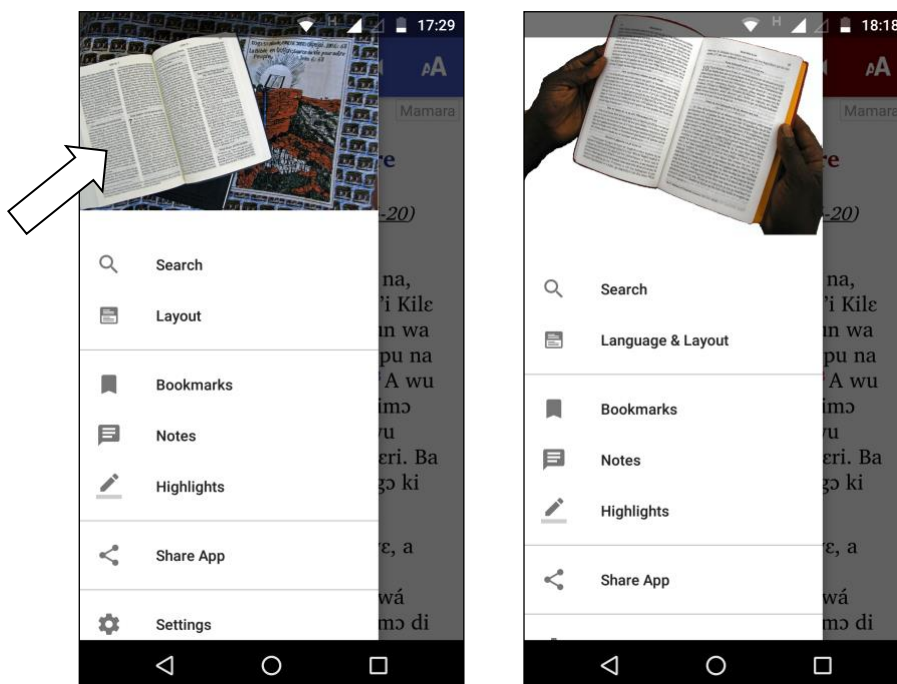
To configure what the user will see within the Layouts configuration dialog:

1. Select the **Books** node for the app in the treeview on the left of the SAB screen.
2. Go to the **Layouts** tab.
3. Choose the default layout and default book collections for each layout type. You can choose to allow the user to select between any of the layout types (Single Pane, Two Pane and Verse-by-Verse) or only one or two of these.



11. Navigation Drawer

You can customise the image that appears at the top of the navigation drawer. It can be a photo, your organisation's logo or any relevant graphic design.



Specify a landscape image file on the **Appearance** ➤ **Graphics** ➤ **Navigation Drawer** page.

12. Contents Menu

When the user launches the app you might want to display a contents menu, giving them an easy way to jump to books and specific references. The contents menu contains a list of images and titles.



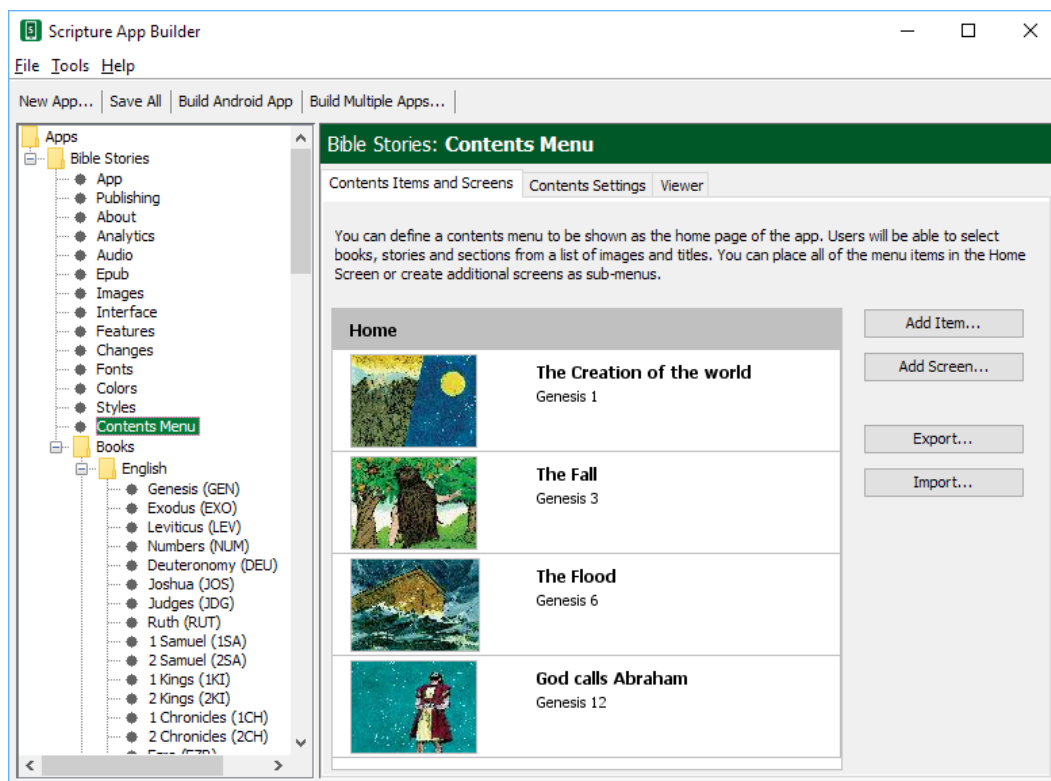
12.1. How do I create a contents menu?

To create a contents menu for your app:

1. Select the **Navigation** ➤ **Contents Menu** page in the Apps tree view on the left of the screen.
2. Ensure that the first tab, **Contents Items and Screens**, is visible.
3. For each contents menu item you want to add, click the **Add Item...** button and follow the wizard to add the item to the screen. You will need to specify:
 - The **target reference**, such as Luke 15 verse 1.

This reference can be a book name, or a book and chapter, or a book chapter and verse. If you have multiple book collections, you can specify an associated layout.

- The **title** of the item, such as 'The Lost Sheep'
 - An optional **subtitle**
 - An **image** (optional) – to appear to the left of the title.
4. You can reorder the items on the screen by selecting rows in the table and dragging up or down.
 5. Go to the **Contents Settings** tab to specify display formatting and navigation options for the menu.
 6. Select the **Viewer** tab to see how the contents menu will appear in the app.



12.2. How do I create additional contents menu screens?

In some cases, you might want to have multiple contents screens, i.e. you select an item in the first screen and it takes you to a second contents screen.

To add a contents screen:

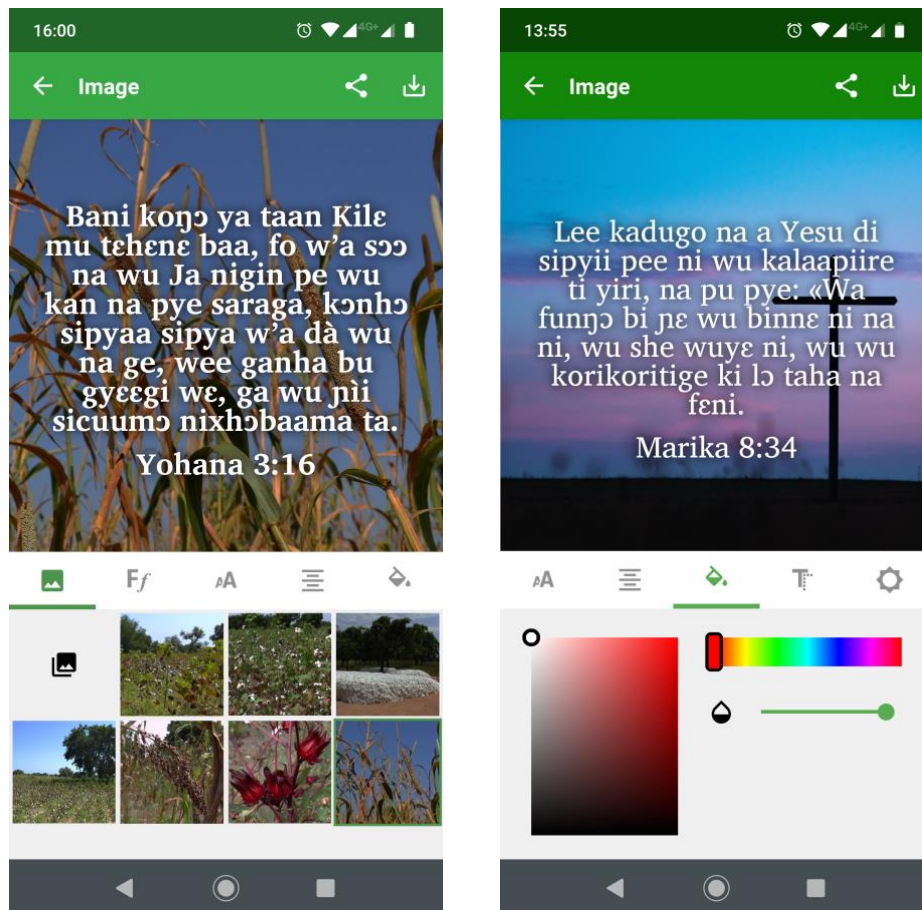
1. Click the **Add Screen...** button and give the screen a name.

2. Click the **Add Item...** button and specify its target to be **Another menu screen** and select the name of your new screen. This will mean that when the user taps on this item in the first screen they will be taken to your new screen.
3. Add menu items to your new screen as described above.
4. You can move items between screens by dragging item rows up or down.

13. Verse on Image

13.1. The Verse on Image editor in the app

If you tap on a verse in the app, it will be selected (underlined) and the text selection toolbar will appear at the bottom of the screen. On this toolbar you can tap the image button which will take you to the Verse on Image editor:









You can change the following **text attributes**:

	Text position (drag the text around the image with your finger)
≡ ↔	Text width (pinch the text or use the slider)

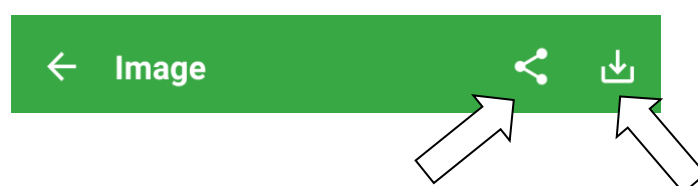
Ff	Text font
Aa	Text size
B	Bold or italics
AV ↔	Letter spacing
↕≡	Line spacing
≡	Left, center or right alignment
◈.	Text color
◐	Text opacity (transparency)
T ⋮	Text shadow or glow

If you double-tap the text, you can **edit the text** itself.

You can change the following **background image attributes**:

	Image selected from one of the built-in images
	Image selected from the photo gallery on your phone or tablet
	Brightness
	Contrast
	Saturation
	Blur

When you are happy with the text and image, use the buttons at the top right of the screen to **save** the image or **share** it with others.



If you have audio and timing files for the selected verse, it is possible to **share the image as a video** with the text and audio combined.

13.2. Configuring Verse on Image in the app builder

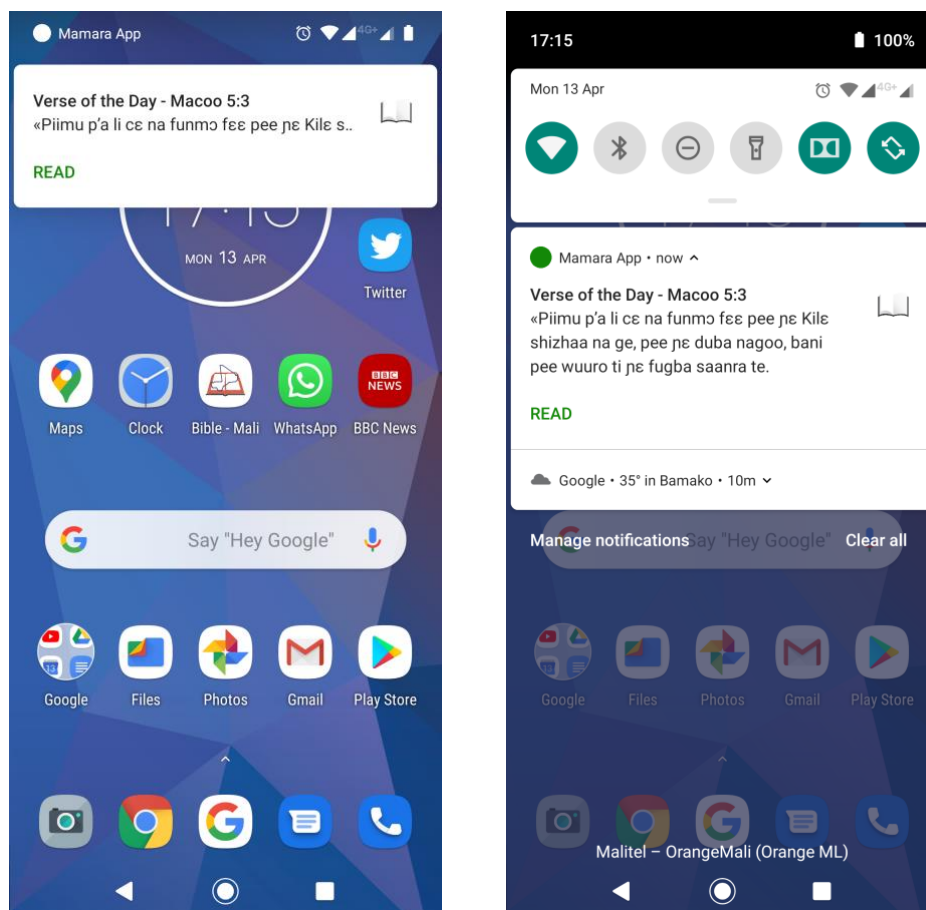
The Verse on Image editor is turned off by default. To enable it in the app builder:

1. Go to the **Engagement** ➤ **Verse on Image** tab.
2. Select **Include the 'Verse on Image' editor in the app**.
3. Select **Allow saving as a video, combining the image and audio**, if you have audio and want to enable the video sharing feature. This will require the image processing library, FFmpeg, to be included within the app.
4. Go to the **Verse on Image** ➤ **Backgrounds** tab.
5. Add some background images. You can get them from:
 - The gallery of images in Scripture App Builder, by clicking the button **Choose from Gallery...**
 - Photos of local scenery that you, friends and colleagues have taken.
 - Images downloaded from websites such as [pexels.com](https://www.pexels.com), unsplash.com and pixabay.com.
6. Go to the **Appearance** ➤ **Fonts** ➤ **Font Files** tab.
7. Add some more fonts if you want to give the user a choice of fonts in the image editor. If you do not specify additional fonts, the font tab will not be displayed in the app.

14. Verse of the Day

14.1. Verse of the Day notifications

Your app can be configured to send a daily verse of the day. The verse appears as a notification in the device's notification tray. When you tap on it, the app will open and highlight the verse.



These notifications work offline. They are sent from the app rather than being sent over the internet. The user can set their preferred time of day for the notification in the app **Settings** screen.

14.2. Configuring Verse of the Day in the app builder

By default, Verse of the Day notifications are not included in the app. To configure them in the app builder:

1. Go to the **Engagement** ➤ **Notifications** ➤ **Verse of the Day** tab.
2. Select **Include Verse of the Day in this app**. If this option is not selected, Verse of the Day will not be included in the app and the app user will not be able to turn it on.

3. If you want the notifications to be turned on by default, select **Turn on Verse of the Day notifications by default**. If this option is not selected, the user will need to go to the Settings screen in the app in order to turn on the notifications.
4. Choose a default **time of day** for the Verse of the Day to appear. The user will be able to change this in the app Settings.
5. Choose the appropriate **Christmas** and **Easter** date traditions.
6. Specify whether the verse can be displayed using the standard system font or whether it needs to be displayed using a custom font. We recommend you select the system font unless you have a more complex script that cannot be displayed well in such fonts.
7. Move to the **Engagement ➤ Notifications ➤ Verses** tab.
8. Review the verses chosen for each day of the year and for the special dates. You can edit individual verses.

15. Highlighting, Notes and Bookmarks

If you tap a verse in the app, it will be selected (underlined) and the text selection toolbar will appear at the bottom of the screen. From this toolbar you can choose to highlight the selected text, add a note, place a bookmark, play audio beginning at this verse, copy the text to the clipboard or share it.

15.1. Highlighting

To highlight one or more verses:

1. Tap on the verse or verses. The app will underline them.
2. Select the highlighter pen button from the text selection toolbar.
3. You will get a choice of highlighter pen colours: yellow, green, orange, blue and pink. Tap on one of them.



To remove a highlight:


1. Tap on the verse or verses for which you want to remove the highlighting.
2. Select the highlighter pen button from the text selection toolbar.
3. Choose the white colour to remove the highlighting.

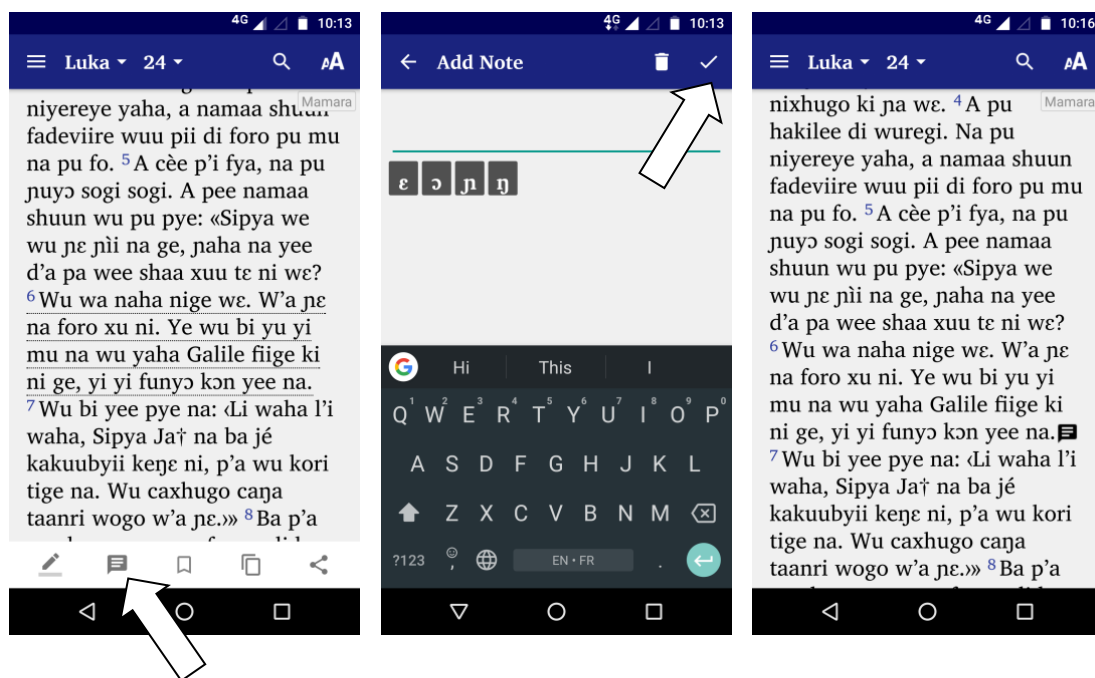
To view a list of highlighted verses, select **Highlights** from the app's navigation drawer menu. On this screen you can sort highlights by date, reference or colour (tap on the Sort button on the app bar). You can delete highlights or tap on an item to go to the highlighted verse. You can also share the list of highlighted verses with someone else (tap on the Share button on the app bar).

15.2. Notes

App users can add personal reflections, teaching notes, feedback on the translation, etc.

To add a note to a verse:

1. Tap on the verse. The app will underline it.
2. Select the Note button from the text selection toolbar.
3. The Add Note screen will appear. Type the note. Special characters can appear as buttons as on the Search screen.
4. Save the note using the tick button on the toolbar. A note icon  will be displayed at the end of the verse. Tap on this icon to see the note in a popup window at the bottom of the screen. Tap on the edit icon to enter the Edit Note screen.




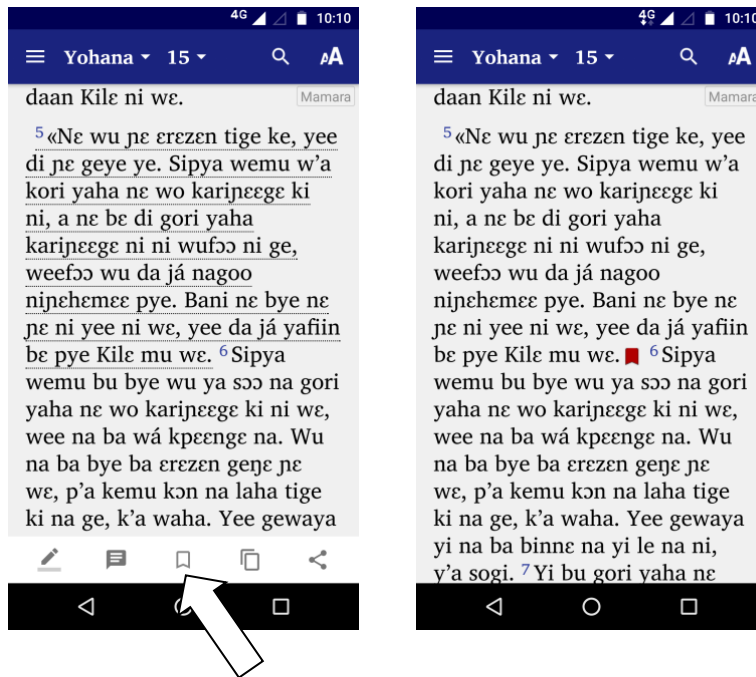
To view a list of notes, select **Notes** from the app's navigation drawer menu. On this screen you can sort notes by date or reference (tap on the Sort button on the app bar). You can edit or delete notes or tap on an item to go to the verse with the note. You can also share the list of notes with someone else (tap on the Share button on the app bar).

You can use the notes sharing feature to allow reviewers in the language community to send feedback on a translation. For this, specify an email address and subject line in the app builder on the **Features** ➤ **Annotations** tab.


15.3. Bookmarks

To add a bookmark to a verse:

1. Tap on the verse. The app will underline it.
2. Select the bookmark button from the text selection toolbar. A red bookmark icon  will be displayed at the end of the selected verse.



To remove a bookmark from a verse:

1. Tap on the bookmark icon  at the end of the verse. The app will underline the verse.
2. Select the red bookmark button from the text selection toolbar. The bookmark will be removed.

To view a list of bookmarks, select **Bookmarks** in the app's navigation drawer menu. On this screen you can sort bookmarks by date or reference (tap on the Sort button on the app bar). You can delete bookmarks or tap on an item to go to the verse with the bookmark. You can also share the list of bookmarked verses with someone else (tap on the Share button on the app bar).

15.4. Configuring highlights, notes and bookmarks

When building your app, you can choose to turn on or off highlights, notes, bookmarks, as well as other actions on the text selection toolbar.

Do this on the **Engagement** ➤ **Annotations** tab.

15.5. Questions about highlights, notes and bookmarks

Here are some questions you might be asking:

Q. Can users highlight phrases in a book introduction or in a non-Scripture book?

- Highlighting, notes and bookmarks are currently limited to chapters with verses. Users cannot highlight titles or sub-headings, just verses of Scripture.

Q. Can users highlight a few words within a verse rather than the whole verse?

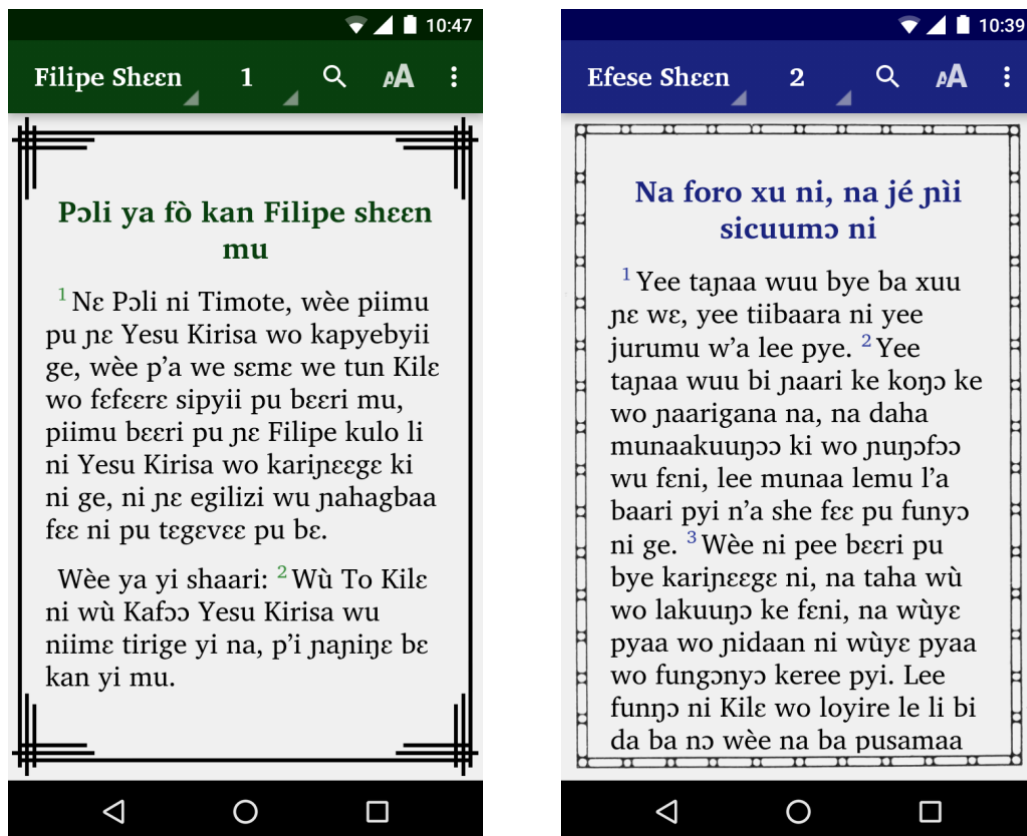
- No, only full verses can be highlighted. A single tap to select a verse provides a simple interface for users and is much easier to do than dragging start and end cursors to select a string of words.

Q. Can users synchronise their highlights, notes and bookmarks with another device?

- Yes, please see the section [User Accounts](#) for instructions on how to set this up.

16. Borders

You can add a decorative border frame around the text in the app, for example:



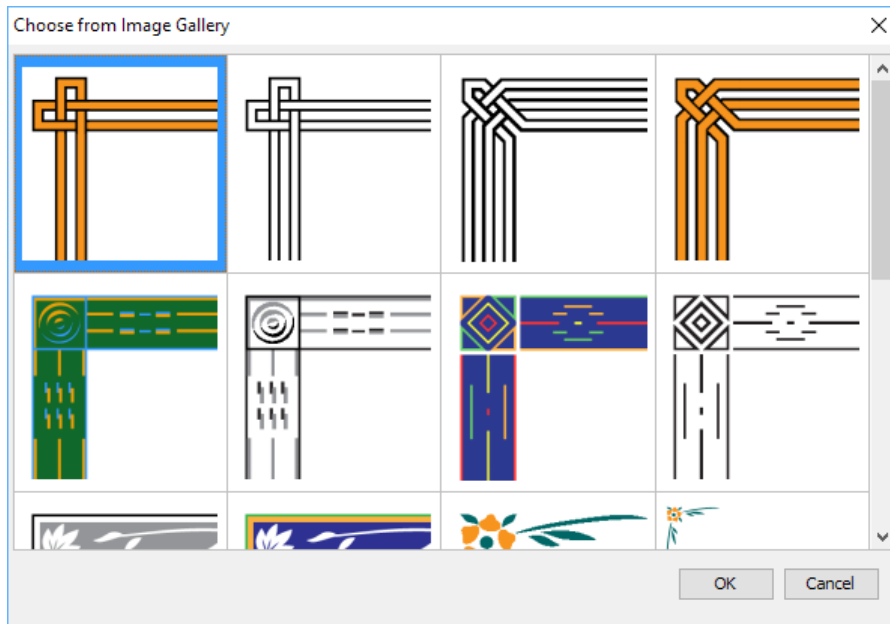
16.1. How do I add a border to the app?

You can select a border from the gallery of border images supplied with Scripture App Builder, or you can create your own border.

To select a border from the border gallery:

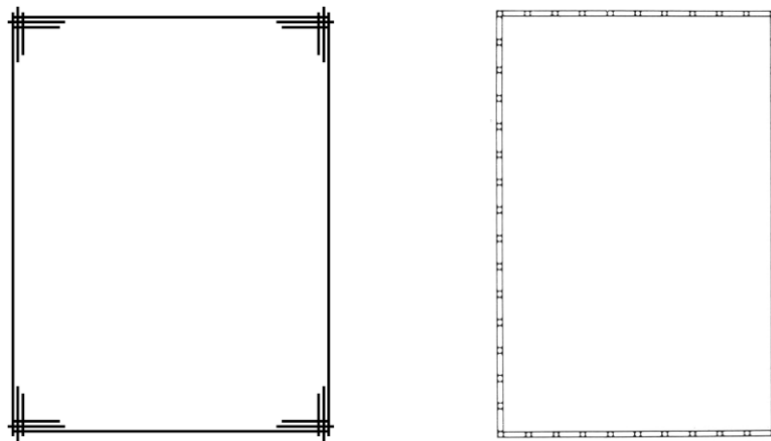
1. Go to the **Appearance** ➤ **Graphics** ➤ **Border** tab.

2. Click **Choose from Gallery...**
3. Select an image from the gallery and click OK.



To add your own border:

1. Create the border as an image file (JPEG or PNG) in portrait orientation. Ensure that the middle rectangular area of the image is blank. This blank area is where the text will be displayed.



You do not need to provide a border image in landscape orientation. This will be generated automatically. It does not matter too much what the width-to-height ratio of the image is, since the app will cut up the border and arrange it for different screen sizes.

Try not to make the border too wide, otherwise there will be limited space to display the text.

2. Go to the **Appearance > Graphics > Border** tab.

3. Click **Add Image...** and select the border image.
4. Ensure that the checkbox **Include a border around the text in this app** is selected.
5. If you want to display the border for some books and not others, or on main chapters but not book introductions, go to the Features page for each book and modify the **Show border around text** and **Show border around introduction** settings.

16.2. How does the app draw the border image?

Scripture App Builder takes the border image file you provide and finds the blank rectangle area inside of it. It then cuts up the remaining image into 8 pieces: top-left corner, top, top-right corner, left, right, bottom-left corner, bottom and bottom-right corner.

When the border is displayed on the app screen, the corner images are scaled according to the screen size, and the side, top and bottom images are scaled and stretched to fill the space between the corners. In this way, the border adapts itself to display with different screen sizes. This technique allows the app to display the border in full screen view, or when there is an audio toolbar displaying at the bottom.

17. Bible Plans

Scripture apps can contain a selection of Bible plans, which include one or more daily readings.

17.1. How do I define a plan?

Plans are defined in text files. Here is the format:

```
\id PLAN1

\title Read through the Gospels
\descr Read the Gospels of Matthew and Mark,
learning about Jesus, what he did, what he said and
why he came.

\img plan1.jpg

\day 1
\ref MAT 1:1-17

\day 2
\ref MAT 1:18-25
\ref MRK 1:1-8

\day 3
\ref MAT 2
```

The standard format markers to use are as follows:

<code>\id</code>	Unique identifier for the plan (for internal use)
<code>\title</code>	Title of the plan (seen by the user)
<code>\descr</code>	Description of the plan (seen by the user)
<code>\img</code>	Image
<code>\day</code>	Day in the plan (\day 1, \day 2, \day 3, etc.)
<code>\ref</code>	Scripture reference to read. There can be one or more \ref lines for each day. References are defined using the Paratext book codes (GEN, EXO, LEV, etc.).

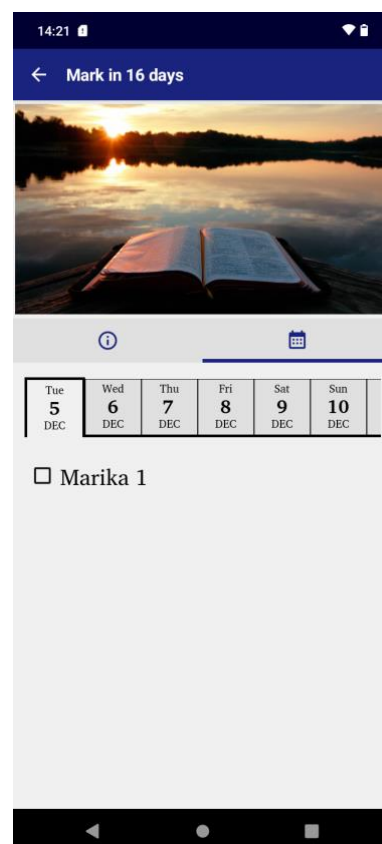
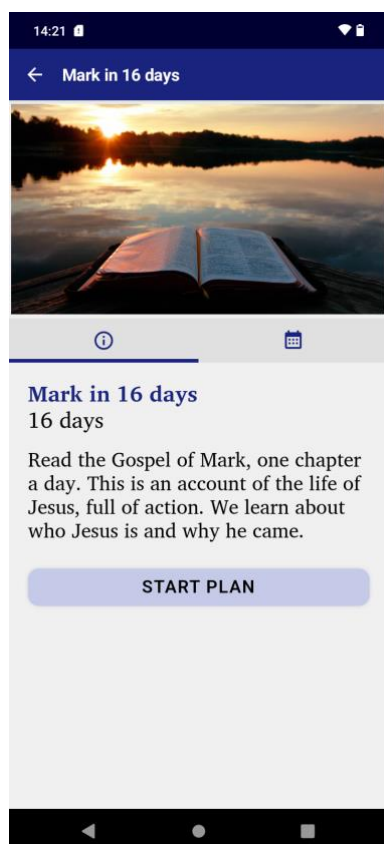
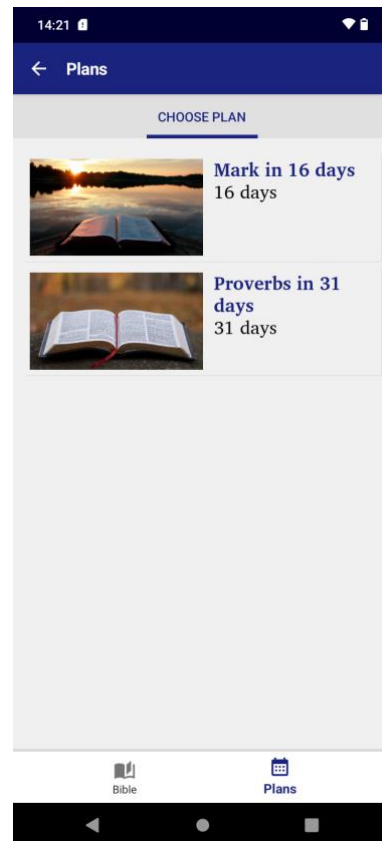
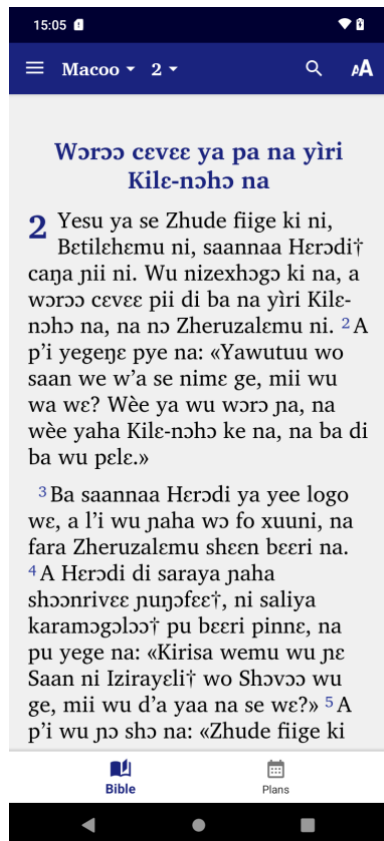
17.2. How do I add a plan to the app?

To add a plan to your app:

1. Go to **Engagement** ➤ **Plans**.
2. Click **Add Plan...**
3. Select a plan definition file (see above on the format of such files) and click **Open**.
4. Double-click the plan in the table to edit the name, description or image.

There is a Plans Gallery with some pre-prepared plans. To use one or more of these, click the **Add Plans from Gallery** button.

Below are some screenshots of the Plans feature in action.



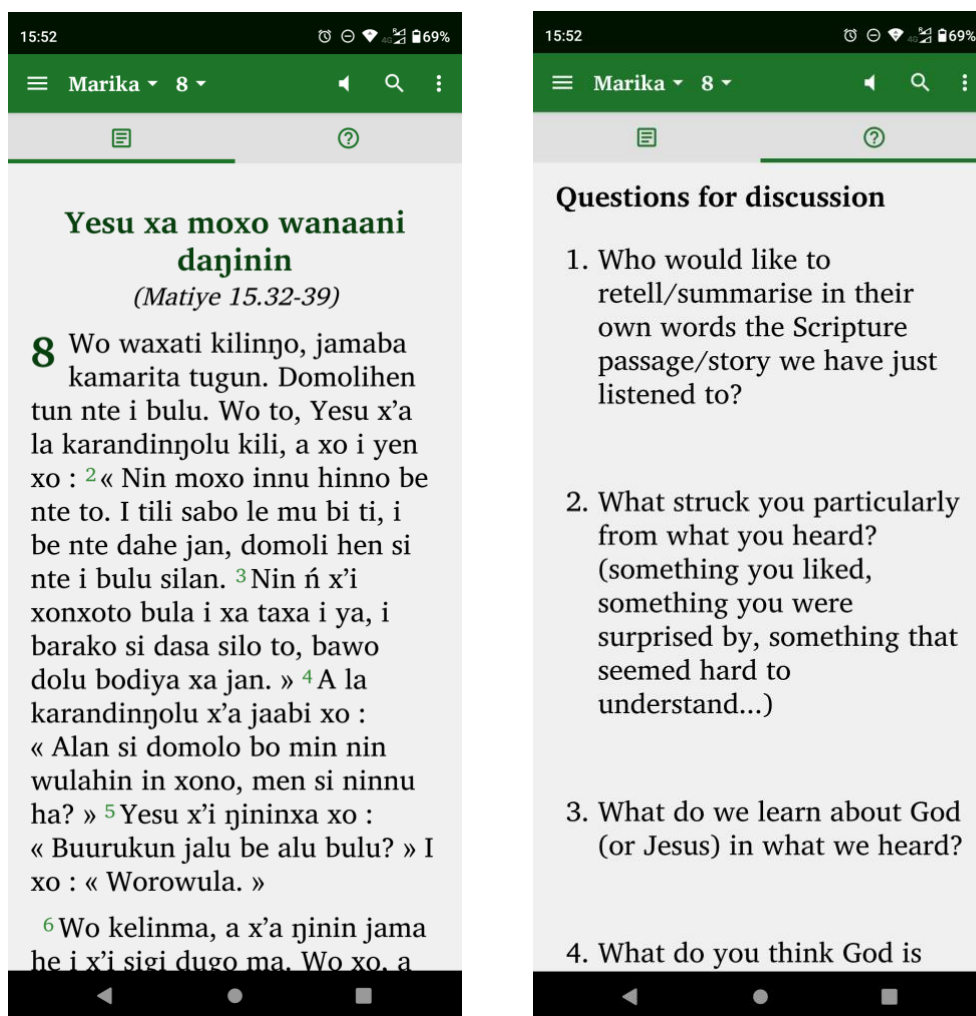
18. Book Tabs

It is possible to create an app which displays books with two or more tabs, for example:

- A “Text” tab in which you display the Scripture text, and a corresponding “Questions” tab which contains Bible study questions.
- A “Text” tab in which you display the Scripture text, and a “Commentary” tab which contains an explanation and application of the text.
- A “Song” tab in which you display the words of a song, and a corresponding “Music” tab which contains the music score or guitar chords.

Tabs can be labelled with a **title** or an **icon**.

You can associate **audio files** with a tab.



Text and Question tabs

To create a tab type:

1. Go to **Navigation** ➤ **Tab Types**.

2. If the type of tab you would like to use is not yet defined, click the Add Tab Type button and follow the instructions in the wizard.

To add a tab to a book:

1. Select the book to which you would like to add the tab, and go to its **Tabs** tab.
2. Click the **Add Tab** button.
3. On the **Tab Type** page of the wizard, select the tab type for the tab you would like to add. Then click **Next**.
4. On the **Book** page of the wizard, click the **Browse** button and select the book file to use in the tab. This could be a text file in SFM format or a Word document.
 - If the book you choose for the tab has a single chapter, this will be displayed in the tab whatever chapter is chosen in the main tab (e.g., the same discovery Bible study questions for each chapter).
 - If the book you choose for the tab has multiple chapters, the app will display the chapter in the tab that corresponds to the chapter selected in the main tab (e.g. a different set of Bible study questions for each chapter).
5. Click **Finish**.
6. If you have any audio files for the tab, go to the **Audio Files** tab and add the audio and timing files.
7. Compile the app and test it on your device.

19. Radio

You can include streaming web radio (internet radio) stations in your app.

To add a web radio station:

1. Go to **Media** ➤ **Radio**.
2. Click **Add Radio Station...**
3. Enter a unique ID for the station (not seen by the user), a short name, and the web address (URL) from which the radio station is being streamed. Sample URLs can be found at <https://streamurl.link>.

Click **Next**.

4. Choose an image to display when the radio station is playing. This could be the radio station's logo.

Click **Next**.

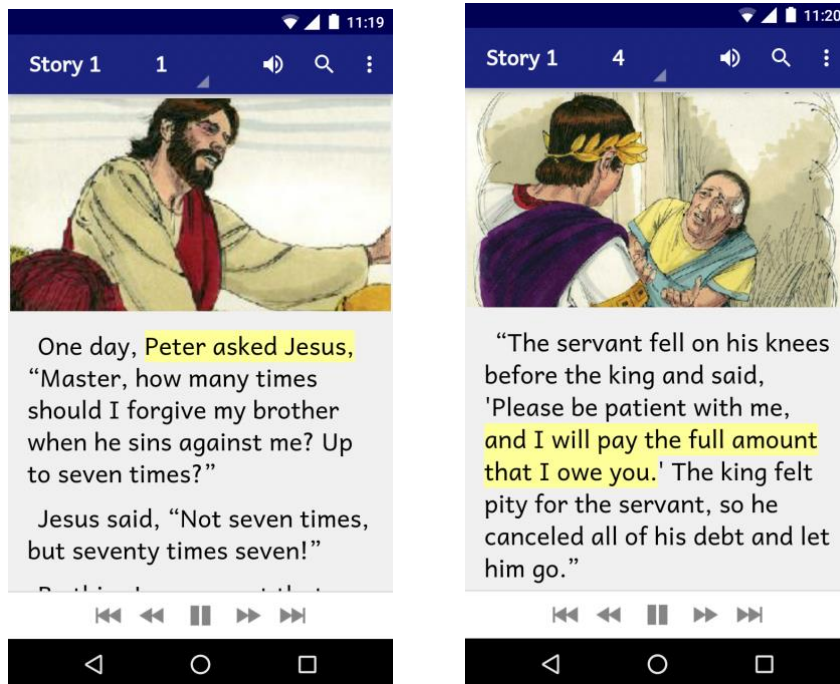
5. Choose whether to add a link to this station from the Contents menu.

Click **Finish**.

6. Compile the app to test the radio stations on your device.

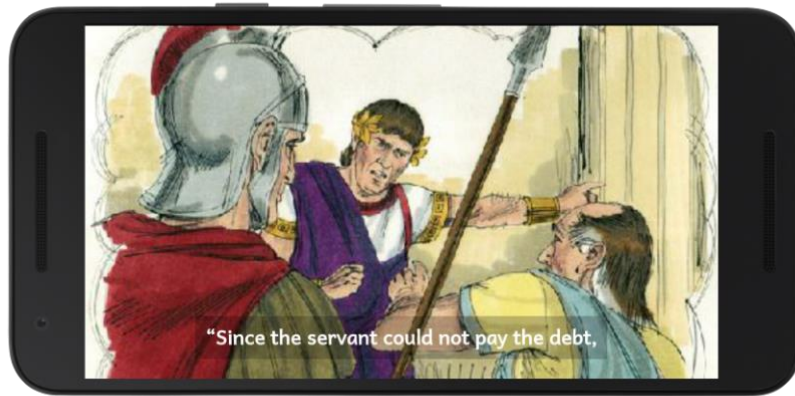
20. Picture Story Books

A picture story book is a book with a picture at the top of each page and a few sentences of text underneath. In the app, the picture will stay at the top of the screen and the text will scroll.



Audio-text highlighting works as it does for Scripture text, and when the audio reaches the end of a page the app will move to the next page of the story.

When the device screen is in landscape orientation, the picture will be displayed to take up the whole screen. If you specify a timing file alongside the audio file, each phrase will be displayed as a subtitle over the image.



You can choose to associate a single audio file with the whole picture story book, or one audio file per page.

20.1. How do I define a picture story book?

Picture story books can be defined in one of two formats:

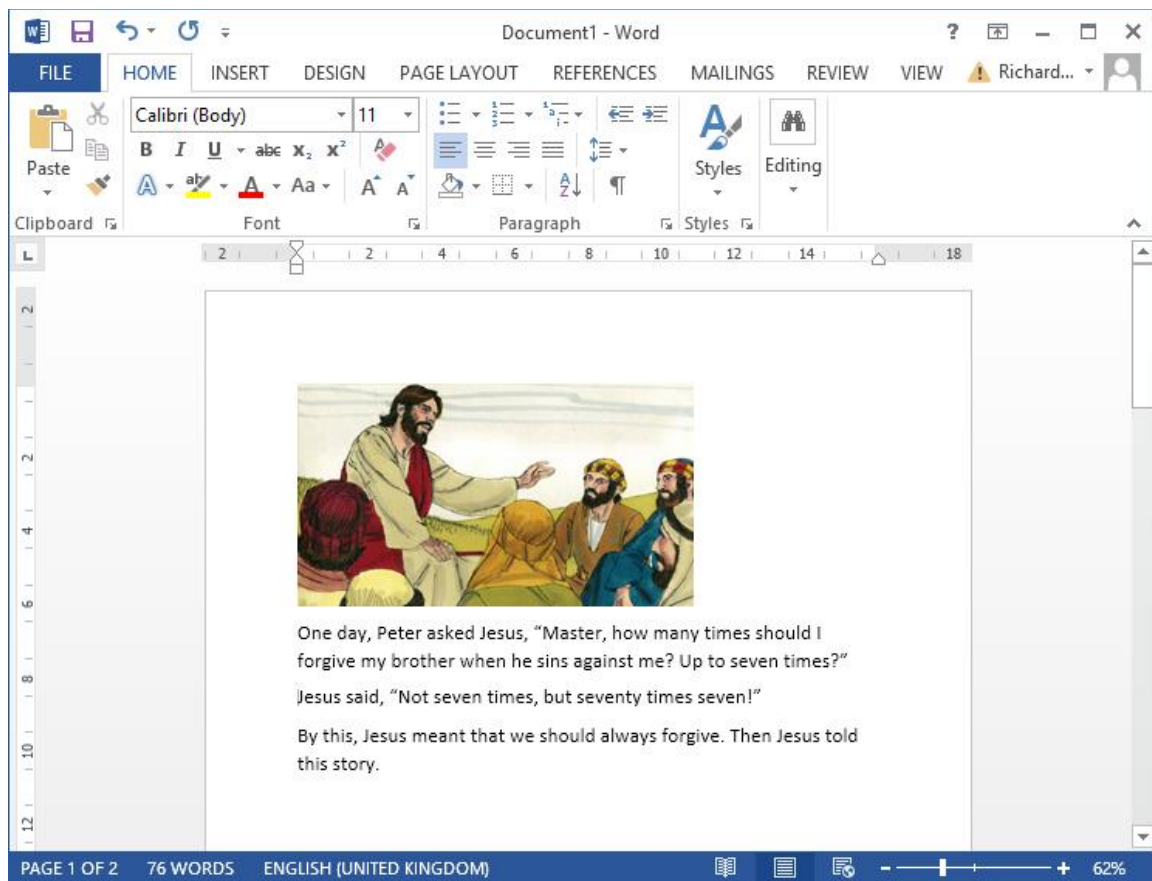
- Microsoft Word (.docx) documents, or
- Text files using standard format markers (SFM).

See 20.2 and 20.3 for a description of these formats.

20.2. What do picture story books in Word documents look like?

To create a Microsoft Word document to define a picture story book:

1. Start with a new blank document in Word.
2. Insert a picture at the top of the first page.
3. Add the story text for this page below the image.

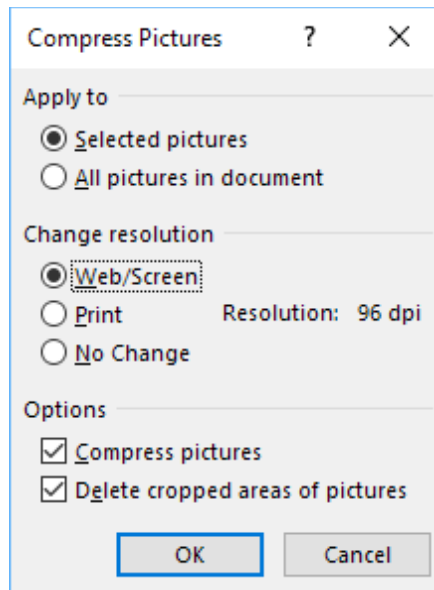


4. Insert a page break after the text using **CTRL+Enter**, or **Insert ➤ Page Break**.
5. Insert a picture at the top of the second page and place the story text under it.
6. Insert a page break after the text using **CTRL+Enter**, or **Insert ➤ Page Break**.
7. Continue adding additional pages in this way.
8. Save the document in **.docx** format.

You can add images to the document in .jpeg, .jpg, .png, .tif or .gif formats and Microsoft Word will convert them automatically to the format recognised by Scripture App Builder.

If you have large image files it is best to resize them. You can do this either before adding them to the document or within Word itself:

1. Select one of the images in Word.
2. Click **Compress Pictures** on the picture format ribbon or toolbar.
3. Select **All Pictures in document**.
4. Select the resolution as **Web/Screen**.



5. Click **OK**.
6. Save the document.

20.3. What do picture story books in SFM format text files look like?

They start with an id of your choice:

`\id STO29`

and a title:

`\toc2 The Story of the Unmerciful Servant`

Each page begins with the `\page` marker:

`\page 1`

followed by an image filename:

`\img obs-29-01.jpg`

and then the text for that page, using USFM markers like `\p` for paragraphs:

`\p One day, Peter asked Jesus, "Master, how many times should I forgive my brother when he sins against me? Up to seven times?"`
`\p Jesus said, "Not seven times, but seventy times seven!"`
`\p By this, Jesus meant that we should always forgive. Then Jesus told this story.`

Here is an example:

\id STO29

\toc2 The Story of the Unmerciful Servant

\page 1

\img obs-29-01.jpg

\p One day, Peter asked Jesus, "Master, how many times should I forgive my brother when he sins against me? Up to seven times?"

\p Jesus said, "Not seven times, but seventy times seven!"

\p By this, Jesus meant that we should always forgive. Then Jesus told this story.

\page 2

\img obs-29-02.jpg

\p Jesus said, "The kingdom of God is like a king who wanted to settle accounts with his servants. One of his servants owed a huge debt worth 200,000 years' wages."

\page 3

\img obs-29-03.jpg

\p "Since the servant could not pay the debt, the king said, 'Sell this man and his family as slaves to make payment on his debt.'"

\page 4

\img obs-29-04.jpg

\p "The servant fell on his knees before the king and said, 'Please be patient with me, and I will pay the full amount that I owe you.' The king felt pity for the servant, so he canceled all of his debt and let him go."

etc.

The book files must be **plain text files**. If you have Unicode characters, the text files should use **UTF-8 encoding**. To create a text file in Windows, use a text editor such as Notepad. To create a text file on a Mac, use TextEdit, remembering to choose Plain text files in the preferences because otherwise the default file type is RTF (which contains a lot of additional formatting codes).

20.4. Where do the pictures go?

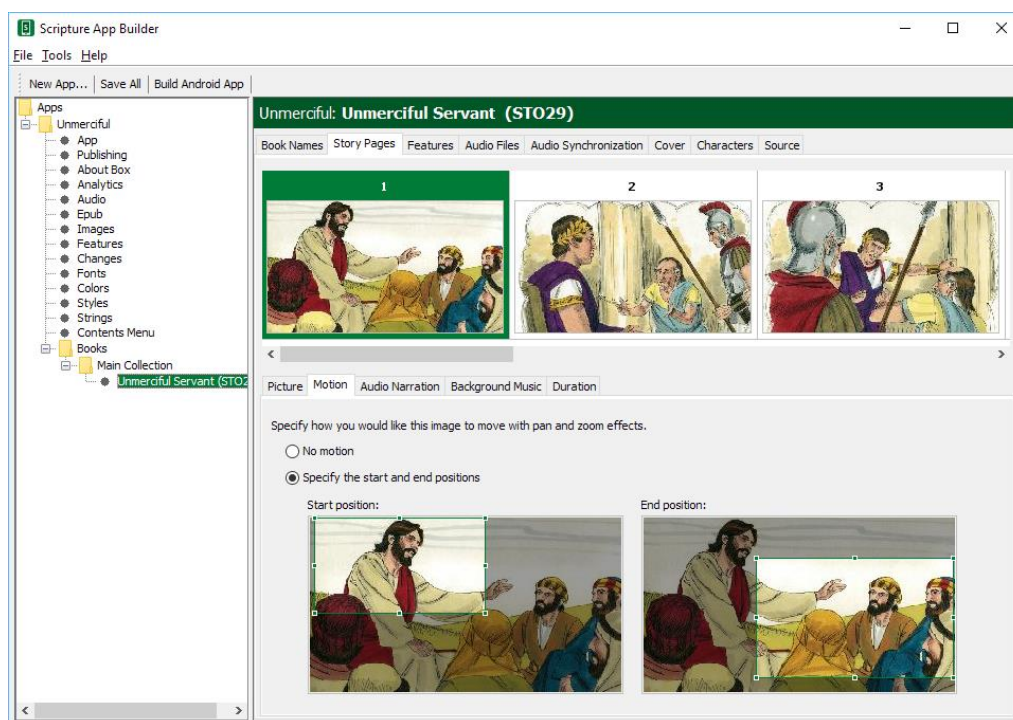
If you have defined the picture story book as a Word document, the images in the document will be added automatically to the app illustrations collection when you add the book. You can see them on the **Illustrations** page for the book.

If you have defined the picture story book as an SFM text file, you need to add the images to Scripture App Builder separately. Do this on the **Media ➤ Illustrations** page. You can use either PNG or JPG files. Try and make them as small as possible without compromising image quality. This will keep your app size small and reduce page load time.

20.5. How can I get the pictures to move when the audio is playing?

You can add pan and zoom effects to the images when the audio is playing. To specify the motion:

1. Go to the **Story Pages** tab for the book.
2. Select an image at the top of the page.
3. Select the **Motion** tab below the images.
4. Select **Specify the start and end positions**.



5. Drag and resize the start and end rectangles in the images to specify the start and end positions.
6. Do this for each image in the book.

20.6. What about font and font size?

A popular font designed for those learning to read is **Andika** or **Andika Compact**.

If you have a small amount of text per picture, it is best to use a larger font size by default. You can change this within Scripture App Builder on the **Appearance** ➤ **Styles** ➤ **Text Styles** page by modifying the **font-size** property of the **body** style.

20.7. What audio timing labels are used?

If you have one audio file for the whole book, timing labels are of the form:

- 1a - 1st phrase on page 1
- 1b - 2nd phrase on page 1

- 1c - 3rd phrase on page 1
- 2a - 1st phrase on page 2
- 2b - 2nd phrase on page 2, etc.

If you have one audio file per page, timing labels are of the form:

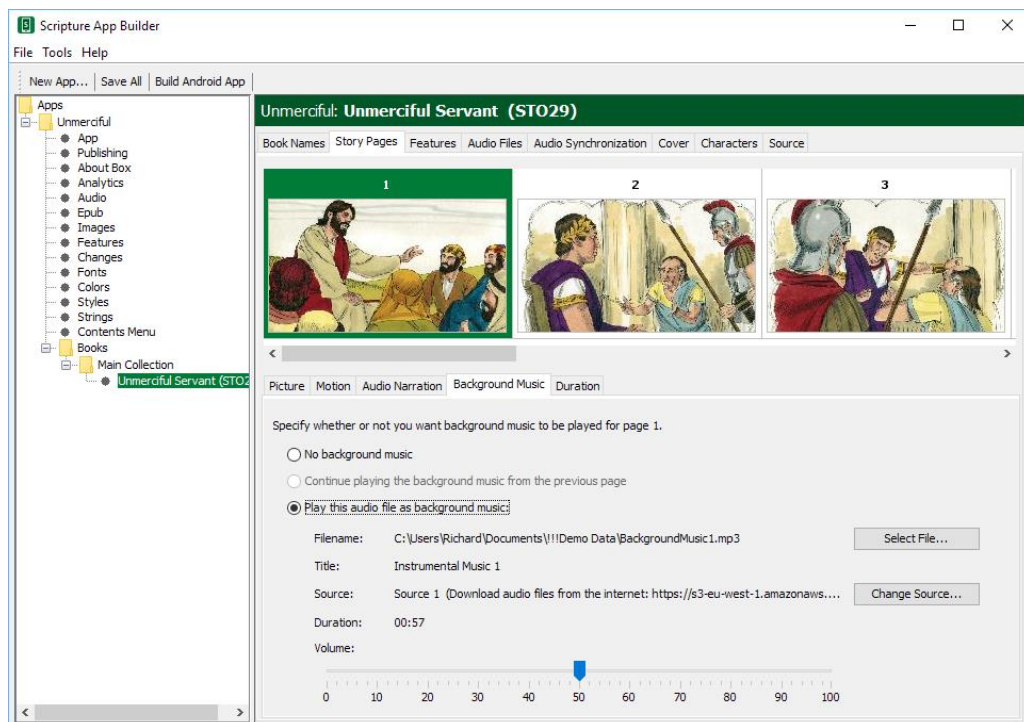
- a - 1st phrase on the page
- b - 2nd phrase on the page
- c - 3rd phrase on the page

To help with the creation of the timing files, generate phrase lists on the **Audio Synchronization** page for the book, or use **aeneas** to automate the synchronization process.

20.8. How can I add background music when the audio is playing?

You can add background music and sound effects behind the audio narration track. To do this:

1. Go to the **Story Pages** tab for the book.
2. Select the first image at the top of the page.
3. Select the **Background Music** tab below the images.
4. Select **Play this audio file as background music**.
5. Click **Select File...** and choose the background music audio file.



6. Drag the slider to modify the volume of the background music.

7. Select each of the following pages in the book and specify whether to Continue playing the background music from the previous page, whether to play a different background music track or whether to stop playing any background music.

20.9. Where can I get sample story book text and pictures?

Have a look at Open Bible Stories here: <https://door43.org/en/obs>

20.10. How can I record the audio files?

Use your favourite sound recording software, such as Audacity. You will need to decide how fast or slow you want the reader to read. In some contexts, it might be good for picture story books to be read fairly slowly.

21. Song Books

21.1. How do I define a song book?

You can build a song book app, with one song per 'chapter'. Song books are defined in one of two formats:

- Microsoft Word (.docx) documents, or
- Text files using standard format markers (SFM).

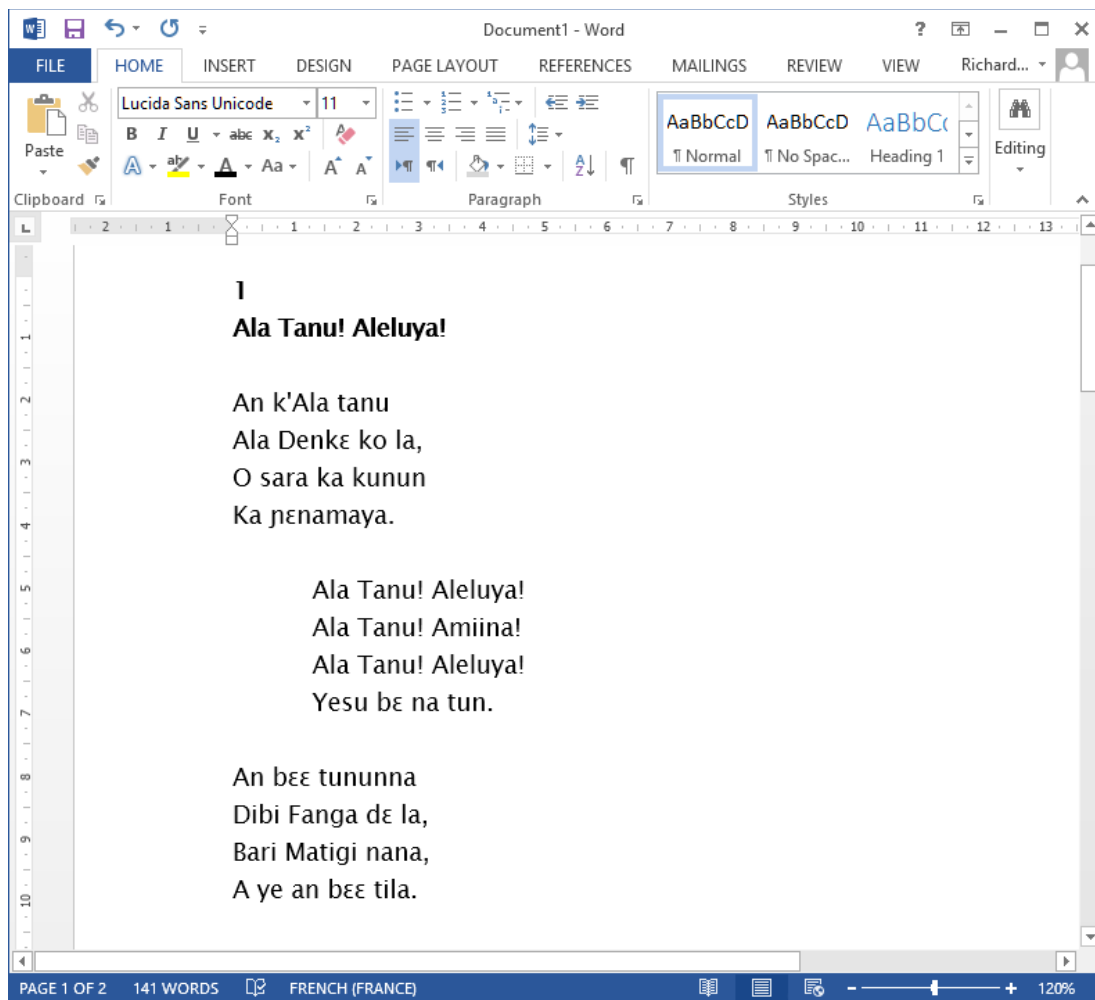
See 21.2 and 21.3 for a description of these formats.

Audio can be added with audio-text synchronization so that each line of the song is highlighted as it is sung.

21.2. What do song books in Word documents look like?

To create a Microsoft Word document to define a song book:

1. Start with a new blank document in Word.
2. Type the songs or copy them from another document.
3. For each song, place the song number on a line by itself before its corresponding title and lyrics.

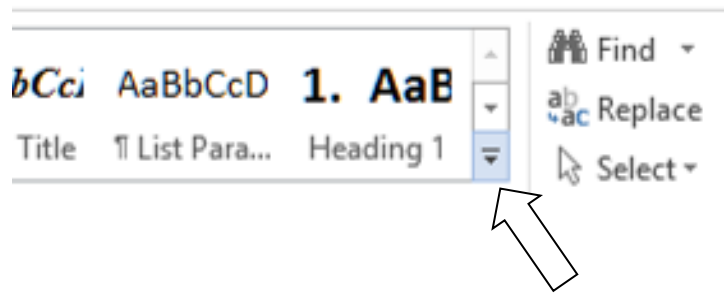


4. Create the following paragraph styles:

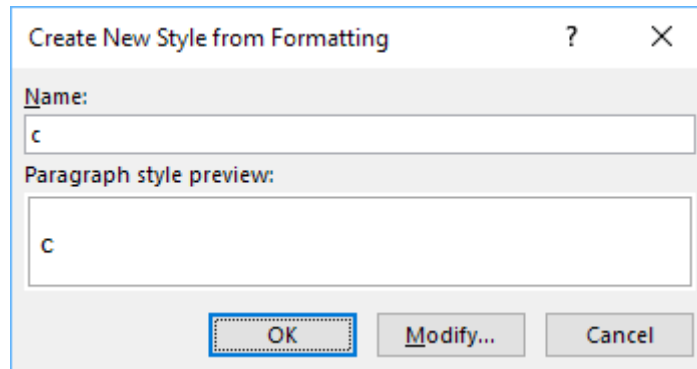
Style	Use
c	Chapter number
s	Song title
q1	Line of the song
q2	Line of a chorus (more indentation)

To create a style:

- Select the line/paragraph of text that corresponds to this style, i.e. if you are creating the style 'c', select one of the lines with a song number. If you are creating the style 'q1', select a line of the song lyrics.
- Click on the down arrow to the right of the styles ribbon.



- c. Then select **Create a style** from the menu that appears.



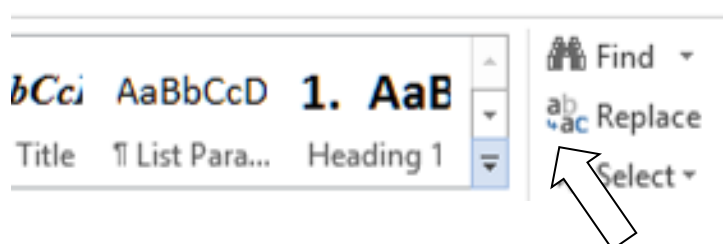
- d. Type the style name into the dialog box and press OK.

Do this for c, s, q1 and q2.

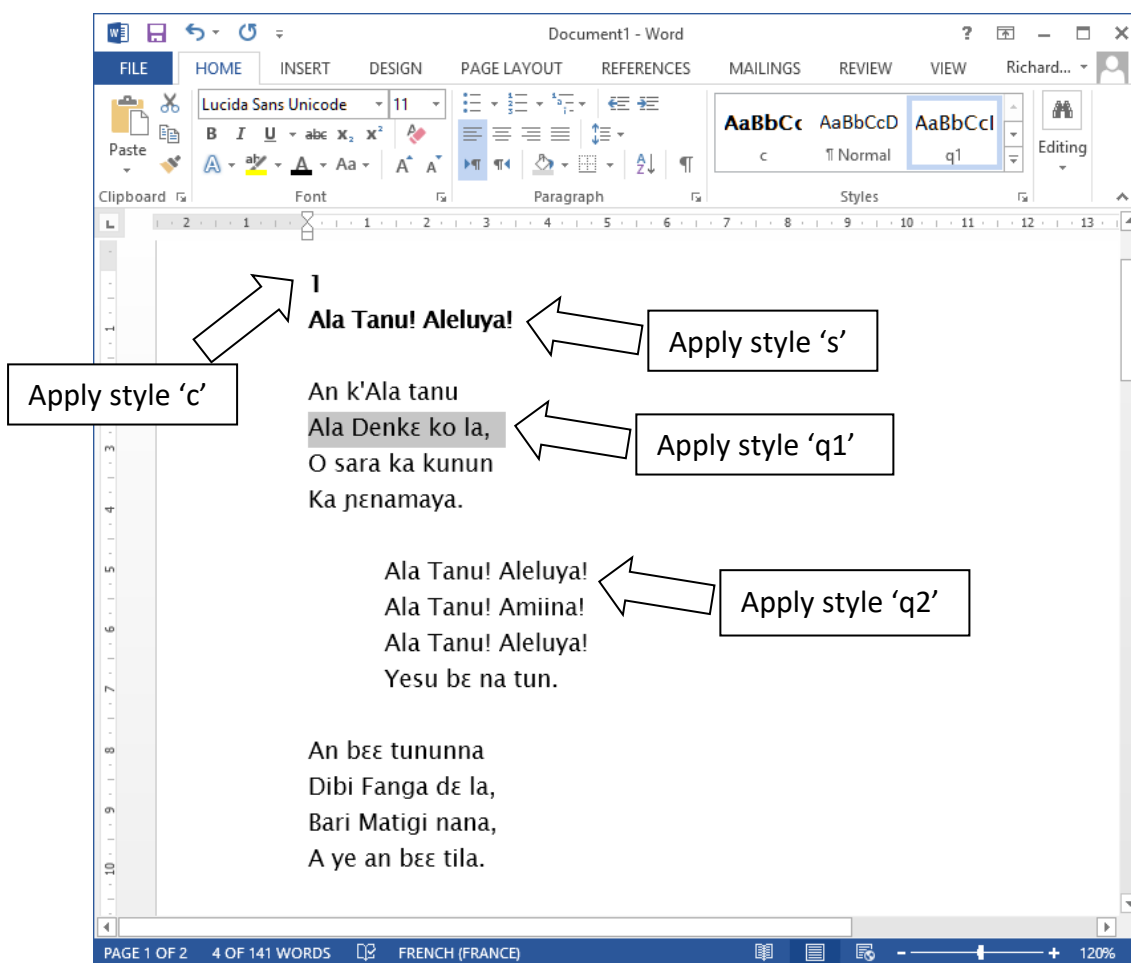
5. Apply the above styles to all the chapter numbers, song titles and the lines of the song lyrics.

To apply a style:

- a. Select the line/paragraph to which you want to apply the style.
- b. Click on the down arrow to the right of the styles ribbon.



- c. Select the style you want to apply.



Tip: The F4 key can be useful as you apply the styles to repeat your previous action.

All of the songs are contained within a single Word document.

6. Save the document in **.docx** format.

21.3. What do song books in SFM format text files look like?

They start with an id of your choice:

\id SONGS

and a title:

\toc2 Our Song Book

Each song begins with the **\c** marker and the number of the song:

\c 1

The marker **\q** (or **\q1**) is used for line of the song lyrics and **\q2** for indented lines, such as for a chorus.

Use **\b** for blank lines between verses.

```
\id BAT
\toc2 Batoli Betiw

\c 1
\s Ala Tanu! Aleluya!

\q An k'Ala tanu
\q Ala Denke ko la,
\q O sara ka kunun
\q Ka jenamaya.
\b
\q2 Ala Tanu! Aleluya!
\q2 Ala Tanu! Amiina!
\q2 Ala Tanu! Aleluya!
\q2 Yesu be na tun.
\b
\q An bee tununna
\q Dibi Fanga de la,
\q Bari Matigi nana,
\q A ye an bee tila.
\b
\q A y'an kunmabo
\q A y'an bee horonya,
\q A y'a ni to an kono
\q K'an dusu jeya.
\b
\q Yesu mana na,
\q A na ke Masa ye,
\q A na dipe bee mara,
\q Ka Sitane tipe.
```

```
\c 2
\s An Fa Ala, Dubatigi

\q An Fa Ala, Dubatigi,
\q An be barika da i ye,
\q Ka i tanu, ka i bato,
\q I ka kan ni bonya ye.
\b
\q2 Kuma duman ke ne da la,
\q2 Ka ne Kisibaa ko fo.
\q2 Aleluya! Aleluya!
\q2 Yesu ye ne kunmabo.
\b
\q Ne be Ala sira taama,
\q An Matigi barika la.
\q Ne y'a don ko Ala sonna
\q Ne ka taa sankolo la
```

If you are used to using Paratext, these SFM codes will be familiar since they are how you mark up poetry in such as Psalms.

All of the songs are contained within a single file.

The SFM songbook file must be a **plain text file**. If you have Unicode characters, the text files should use **UTF-8 encoding**. To create a text file in Windows, use Notepad. To create a text file on a Mac, use TextEdit, remembering to choose Plain text files in the preferences because otherwise the default file type is RTF (which contains a lot of additional formatting codes).

21.4. How can we associate audio with each song?

A separate audio file and corresponding timing file needs to be created for each song.

Timing labels are of the form:

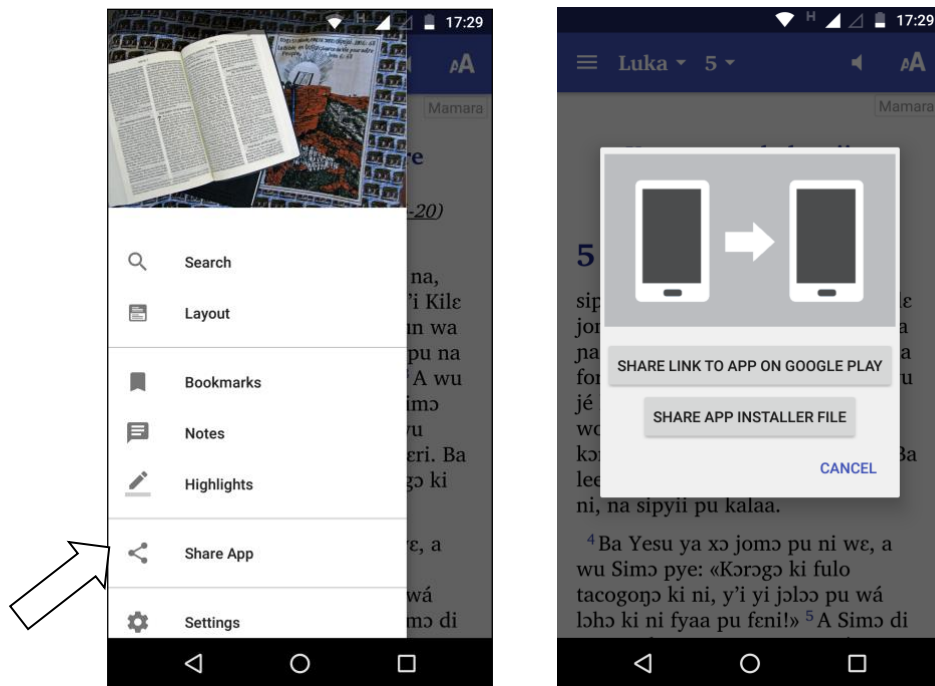
- a - 1st line of the song
- b - 2nd line of the song
- c - 3rd line of the song
- d - 4th line of the song, etc.

To help with the creation of the timing files, generate phrase lists on the **Audio Synchronization** page for the book.

Automated synchronization is unlikely to work for songs, so you will need to use the manual tagging method using Audacity and Timing Labels Assistant. Please see the document *Using Audacity for Audio Text Synchronization* for instructions.

22. Sharing Apps

A **Share App** item can be added to the app navigation drawer menu to enable users to share the app with others. You can configure this on the **Navigation Drawer > Navigation Drawer > Sharing** page.



There are two ways of sharing the app. You can choose to allow the user to use one or both of them:

- **Share Link to App on Google Play:** this option allows the user to send someone the link to the app on the Google Play store, e.g. by email or social media. You should enable this only if the app has been published on Google Play.
- **Share App Installer File:** this option allows the user to share the APK file with another device, e.g. by Bluetooth, Wifi transfer or email. This option is especially useful in contexts where not everyone has easy internet access and where you want to promote offline app sharing.

23. Deep Linking

Deep Linking allows you to direct people to specific pages within the app, using a link on a website, in a messaging app, social media or email.

A deep link can look something like this:

mamara-bible://LUK.15

or

<https://mamara.org/bible/LUK.15>

or

<https://www.wycliffe.org/mexico/huichol/bible?ref=LUK.15>

If you share one of these links with a friend on WhatsApp, Facebook or email, and they tap on the link on their smartphone, the app (if already installed) will open and go directly to Luke chapter 15.

23.1. Setting up Deep Linking

To set up deep linking in your app:

1. Go to the **App** ➤ **Deep Linking** page within Scripture App Builder.
2. Select **Enable Deep Linking**.
3. Specify one or two link **schemes** to be associated with your app: a custom URI and/or a URL.

Custom URI Scheme

The first type of link scheme is a **custom URI** (Uniform Resource Identifier). This is a sequence of characters beginning with a letter and then any combination of letters, digits, plus (+), period (.), or hyphen (-). It is followed by a colon and two forward slashes (://).

Here are some examples:

mamara-bible://
biblemalinke://
proverbstoday://
tharaka-scripture://
gospels-tanzania://

Make sure that you do not choose a custom URI already in use by a popular app, such as:

fb://	Facebook
twitter://	Twitter
whatsapp://	WhatsApp
youtube://	YouTube
youversion://	YouVersion Bible App

URL Scheme

The second type of scheme is a **URL** (Uniform Resource Locator), which is like a web address. It begins with <http://> or <https://>. It can include a path.

Here are some examples:

<https://mamara.org/>
<https://mamara.org/bible/>
<http://www.gospels-tanzania.org/>
<https://wycliffe.org/mexico/huichol/bible/>
<https://scriptureearth.org/mexico/hch/>

23.2. Creating Deep Links

This section explains how you create deep links to content in the app. We will look at how to specify Bible references (books, chapters, verses), book collections, and how to turn the audio toolbar on when the app opens.

Specifying Bible references in a deep link

A reference is specified in the form BBB.C.V, where BBB is the book ID, C is the chapter number and V is an optional verse number.

Examples:

LUK.15	Luke chapter 15
JHN.3.16	John chapter 3 verse 16
EPH.2.8	Ephesians chapter 2 verse 8

Please note that:

- The reference is case insensitive, so luk.15, Luk.15 and LUK.15 are all recognised as referring to Luke chapter 15.
- The app will recognise [OSIS book abbreviations](#) as well as Paratext codes, such as Luke.15, 1Thess.3.5, 1Pet.2.

There are two ways of specifying the reference in a deep link:

Method 1: Append the reference immediately after the link scheme, e.g.

mamara-bible://JHN.3.16
<https://mamara.org/bible/JHN.3.16>

Method 2: Specify the reference in a query string, after '?' and 'ref=', e.g.

mamara-bible://?ref=JHN.3.16
<https://mamara.org/bible?ref=JHN.3.16>

When someone taps on this link, the app (if installed) will open at the specified book, chapter and verse, using the currently selected layout and book collections.

Specifying book collections in a deep link

You can specify a book collection in the deep link, using a reference of the form DDD/BBB.C.V, where DDD is the book collection ID.

Here are some examples:

```
mamara-bible://MYK/JHN.3.16  
https://mamara.org/bible/MYK/JHN.3.16  
https://mamara.org/bible?ref=MYK/JHN.3.16
```

If you specify a book collection, the app will ensure that this book collection is being displayed:

- If the app is in single pane layout, the book collection will be changed to the specified collection if it is not there already.
- If the app is in two pane or verse-by-verse layout, and if the specified book collection is not among the two or three on the screen, the layout will be changed to single pane with the specified book collection.

Turning on the audio toolbar in a deep link

You can turn on the audio toolbar when the app launches, with the addition of the **audio=1** parameter in the query string of the deep link.

Here are some examples:

```
mamara-bible://JHN.3.16?audio=1  
https://mamara.org/bible/MYK/JHN.3.16?audio=1
```

Note that a query string begins with the question mark (?) character before the first parameters, and then an ampersand (&) before subsequent parameters. So if you are using the 'ref=' method of specifying a reference, you will need to use '&', e.g.

```
mamara-bible://?ref=JHN.3.16&audio=1  
https://mamara.org/bible?ref=JHN.3.16&audio=1
```

23.3. Deferred Deep Linking

The deep links described above will only work if the user already has the app installed. If they do not have the app, when they tap on the link, their device will not recognise it and an error message will appear.

Branch (branch.io) provides a way of handling **deferred deep links**. If a user taps on a link to the app but does not have the app installed yet, the link will first redirect the user to the app store to download the app. Then after the app is installed and launched for the first time, the user will be taken to the deep link page.

Setting up the app to use Branch

To use Branch for deferred links:

1. Go to the **Branch** website, <https://branch.io/> and **Sign Up** for an account if you do not already have one.
2. Create a new app in the Branch dashboard.
3. Go to the **Link Settings** page, find the **Android** section and click the checkbox 'I have an Android App'.
4. Enter the custom URI scheme and where to download the app (e.g. on Google Play).
5. Within Scripture App Builder, go to the **App** ➤ **Deep Linking** page.
6. Select **Use Branch for Deferred links**.
7. Enter the **Branch Key** for this app. This can be found on the **Account Settings** page in Branch and will be something like:

key_live_abCdEF2GQ7ID9rLPlxk91khcZmpGZI71

8. Enter the **Default Link Domain** for this app. This can be found on the **Link Settings** page in Branch and will be something like:

mamara-bible.app.link

Creating deferred links

Deep links using Branch will look something like this:

<https://mamara-bible.app.link?ref=JHN.3.16>

where mamara-bible.app.link is the **Default Link Domain** specified on the Branch **Link Settings** page.

Please note that you need to use the **?ref=** method of specifying the reference. You cannot use <https://mamara-bible.app.link/JHN.3.16>.

24. Analytics

If you enable Analytics, the app will connect to the internet from time to time to send app usage information to one or more analytics accounts. This will give you an idea of the extent to which people are interacting with the app.

The information sent will include the model of the device (such as 'Google Nexus 7', 'Samsung Galaxy S4'), the Android version (such as '4.2'), the mobile network provider and an approximate location (city/country). No personal information is included.

You can configure your app to send usage data to one or more of the following analytics engines:

Firebase Analytics	Sends data to a Google Firebase Analytics account of your choice.
Amplitude Analytics	Sends data to an Amplitude account of your choice.
S3 Digest Analytics	Sends a digest of analytics data to an Amazon S3 Bucket of your choice.
FCBH Digest Analytics	Sends a digest of analytics data to Faith Comes By Hearing (FCBH).

To set up analytics:

1. Go to the **Data & Analytics** ➤ **Analytics** page for the app.
2. Select **Enable Analytics**.
3. Click **Add Analytics Account...**
4. Choose an account type and enter your analytics account information.
 - For Firebase Analytics, you will need a google-services.json configuration file for your account.
 - For Amplitude Analytics, you will need an API Key.
 - For S3 Digest Analytics, you will need an S3 Bucket ID and an Identity Pool ID.
 - For FCBH Analytics, you do not need any additional ID information.

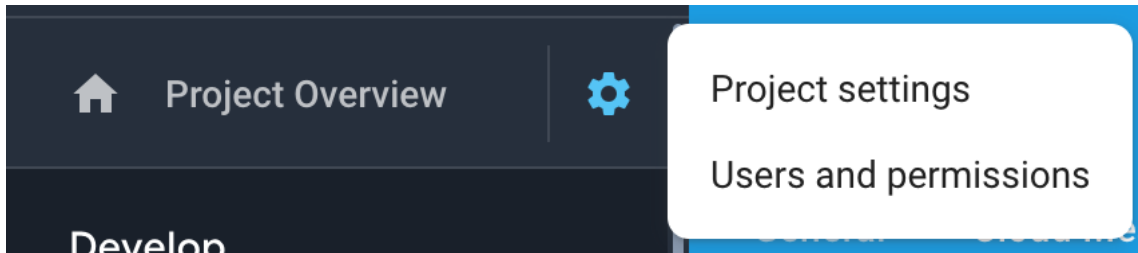
24.1. Firebase Analytics

To sign up for Firebase Analytics, ensure you have a Google account.

You will need to:

1. Go to the Google Firebase website at <https://firebase.google.com/> and ensure you are signed in with your Google account.
2. Click **Add project** to create a Firebase project for this app.
3. In Step 1, you will need to give your new project a **Project name**.
4. In Step 2, titled **Google Analytics for your Firebase project**, select **Set up Google Analytics for my project** and press **Continue**.

5. In Step 3, titled **Configure Google Analytics**, click on the drop-down box and choose **Create a new account**. Give it a name, which can be the same as your Firebase project name.
6. Check the boxes to confirm that you accept the analytics and data protection terms. Click **Create project** and wait a few seconds for the project to be created.
7. Click the Settings button (a cog wheel icon near the top) and select **Project settings**.



8. On the **General** tab of the Settings, scroll down to the **My apps** section and click the Android app icon.
9. On the **Add Firebase to your Android app** page, enter your app package name and click **Register app**.
10. Download the config file, **google-services.json**. That is all you need from the app registration. You can ignore the information on the rest of the screens and return to Settings.
11. Go to the **Data & Analytics** ➤ **Firebase** page in Scripture App Builder.
12. Select **Firebase Analytics** as one of the features to use in the app.
13. On the **Firebase Configuration** ➤ **Android** tab, click the **Browse** button and find the **google-services.json** config file that you have just downloaded from the Firebase console.

24.2. Amplitude Analytics

To use Amplitude analytics, you will need to create an account. Go to:

<https://amplitude.com>

You will need to:

1. Click **Sign Up** at the top right of the screen and create an account. You will receive an email to finish activating your account. Copy the link to your browser and go to the page and complete the activation process.
2. You will be prompted to create a new organization. You can do that to invite team members to join your organization and access the data. You will also be prompted for some additional information.
3. Click **Create Project**, enter the project name and click **Create**. There will be a project for each individual app.

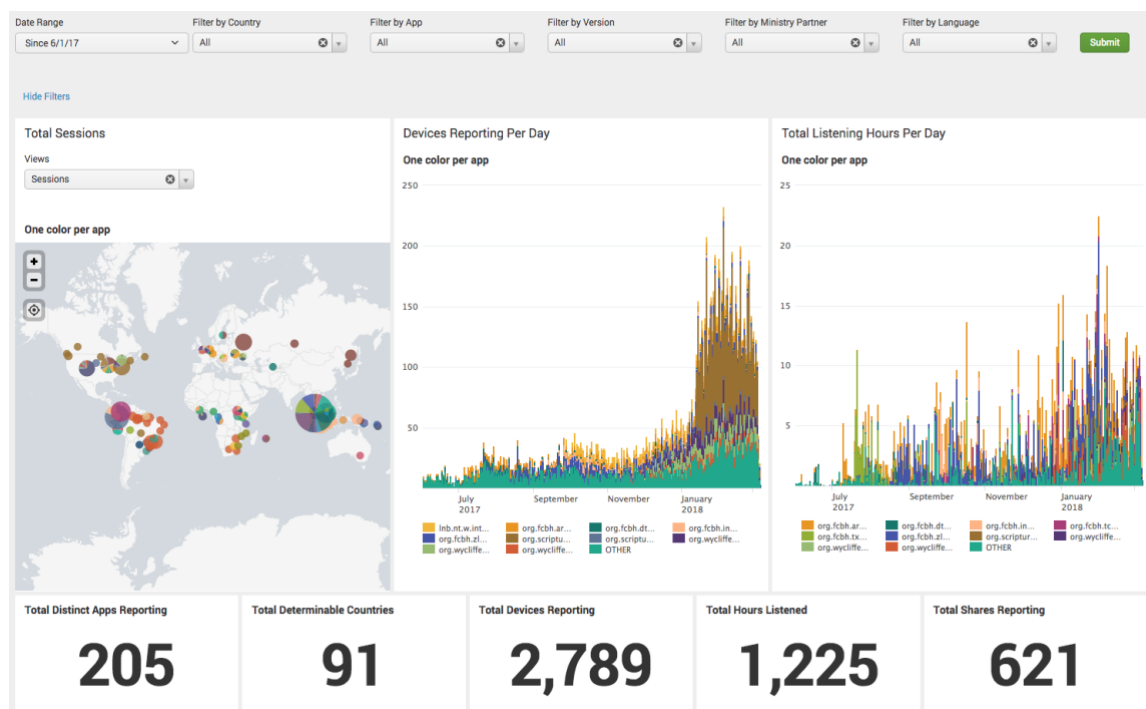
- Click **Projects** on the left of the screen. This will show the list of projects and the properties including a long string of hexadecimal characters under the label **API Key**.
- Highlight and copy this string into the API Key field in Scripture App Builder.

24.3. FCBH Digest Analytics

Faith Comes By Hearing (FCBH) has established an analytics service for use with Scripture App Builder. By selecting FCBH Digest Analytics as the Account Type, your app will send a small daily digest (about 300 bytes for each day the app is used) of compressed, completely anonymous data (no personal, phone, or GPS location information) via an encrypted transport (https) to an Amazon data center in Asia, when the device is connected to the Internet (via cell or WIFI) while the app is running.

FCBH uses the resulting data for two purposes: 1) They make semi-annual contributions to text-holder organizations, proportional to the usage of corresponding audio recordings, and 2) They show usage data to visitors and donors to encourage and inspire involvement. Access to this service (data and dashboards) is available to interested partners (e.g. app builders) so they can easily track usage. Send an email to sab-analytics@fcbhmail.org to request access.

Data is aggregated by an FCBH server and analyzed using Splunk. A snapshot of one dashboard is below. The dashboard is interactive and allows drill-down to specific apps or versions or countries. The dashboard is a work in progress and interested ministry partners are welcome to contribute or build their own dashboards on the Splunk server.



While no location information is uploaded, Amazon's servers record the IP addresses where data uploads occur from. FCBH then uses [Maxmind's](#) free database to estimate

device locations (at approximately 99% and 95% accuracy for country and city respectively).

24.4. S3 Digest Analytics

S3 Digest Analytics uses the same mechanism as FCBH Digest Analytics to report a small digest of analytics data. However, you are responsible for hosting the Amazon S3 Bucket and processing the received data. Files are uploaded to the S3 Bucket and in a sub-folder based on the format (e.g. the current format is f1) and stored with a unique filename using a randomly generated [GUID](#) as the basename. We are working with FCBH to package a Splunk configuration so that you can deploy your own server to analyze the data.

To use S3 Digest Analytics, ensure you have admin permissions to an Amazon AWS account, and go to:

<https://aws.amazon.com/console/>

You will need to:

1. Click **Sign In to the Console** at the top right of the screen.
2. Create an S3 Bucket.
 - a. Go to the **S3** Service and click **Create bucket**, enter a Bucket name, select a Region near where the app will be distributed, and click **Next**.
 - b. On the **Set properties** and **Set permissions** steps, use the defaults and click **Next**.
 - c. Review the configuration and click **Create bucket**.
 - d. Copy the **Bucket name** into the **S3 Bucket ID** field in Scripture App Builder.
3. Create a Federated Identity.
 - a. Go to the **Cognito** Service and click **Manage Federated Identities**. The first time you use this service it will start creating an identity pool for you. If the AWS account already has identity pools, then it will show a grid of existing one. If this is the case then click **Create new identity pool**.
 - b. Enter an **Identity pool name**, click **Enable access to unauthenticated identities** (which allows users of the app to submit analytics without logging into some service), and click **Create**.
 - c. Click **Show Details** to see the **Role Name** for the unauthenticated identities (in the next step, we will give them permission to put objects in the bucket created in the previous step). Click **Allow**.
 - d. Copy the **Identity pool ID** value inside the quotes in the **Get AWS Credentials** section of the **Sample code** page shown after completion of the previous step. Copy this string into the **Identity Pool ID** field in Scripture App Builder.

4. Give permission to put data into the S3 Bucket.
 - a. Go to the **IAM** Service and click **Roles** category.
 - b. Click on the Role created in step #3 (e.g. Cognito_<IdentityPoolName>Unauth_Role).
 - c. Click **Add inline policy**, click **Service** and choose **S3**.
 - d. Click **Actions**, type in *PutObject* to search for actions, and check **PutObject** to select that action.
 - e. Click **Resources**, use default Specific, click on **Add ARN** link, enter the **Bucket name** from step #1 into the **Bucket name** field, click the **Any** checkbox at the end of the **Object name** field, and click **Add**.
 - f. Click **Review policy**, enter a name in the **Name** field, and click **Create policy**.

24.5. Data Payload for FCBH and S3 Digest Analytics

Both FCBH Digest Analytics and S3 Digest Analytics use the same JSON payload format. A complete sample data payload is below:

```
{ "startTime": "20180306T0835Z", "period": 1440, "id": "12db7e3f-93d9-4370-b12b-fe048804e4f5", "package": "org.fcbh.hndwtn2", "version_name": "1.0.1", "sessions": 1, "sessionMins": 21, "shares": 3, "dams": [ { "damid": "HNDWTCN2DA", "playMins": 25, "chaps": [ "MRK 1", "MRK 1", "MRK 2", "MRK 2", "MRK 3", "MRK 4", "MRK 4" ] } ] }
```

The payload is 296 bytes uncompressed and 224 bytes compressed. This sample is comprised of the following fields:

- One day of activity (1440 minutes), starting on 2018-03-06.
- The id is a [GUID](#) which was randomly generated on the phone when the app was first launched, enabling determination of how many unique installations of the app are in use (but no user-identifying information).
- Package and version indicate which app is in use.
- This report was for a single 21-minute session. This (and other) values would be incremented if the app had been used multiple times within the reporting period.
- The user pressed Share in the app 3 times.
- The app played FCBH audio HNDWTCN2DA, which is a **Fileset ID** from FCBH's [Bible Brain](#).
- The app played audio for 25 minutes. Since this is greater than 21 minute session time, the audio must have been playing while the screen was off for 4 minutes. Comparison of these fields enable some rough comparison of audio-only use, versus use of audio in combination with text.
- Mark 1-4 were played. Mark 1 and 4 being listed twice each indicates that the user pressed play twice in each chapter (and thus, also pressed stop in each).

25. Push Notifications

You can send direct messages over the internet to all the users who have your app installed on their phone.

Such messages could include:

- Sending greetings on special occasions.
- Sharing news of app updates, new resources and special offers.
- Encouraging people to use the app.

25.1. Setting up Firebase Cloud Messaging

To set up the push notifications feature:

1. If you have not already done so, follow the instructions on setting up a **Firebase project**, described in the section above on [Firebase Analytics](#).
2. In Scripture App Builder, go to the **Data & Analytics** ➤ **Firebase** tab and enable **Firebase Cloud Messaging**.

25.2. Sending messages

To send a push notification message to all users of your app:

1. Go to the **Firebase Console** for your Firebase project.
2. Select **Cloud Messaging** from the menu on the left of the screen.
3. Click **Send your first message**.
4. Enter the **Notification title** and **Notification text**, then click **Next**.
5. In the **Target** section, you want to send messages to users of your app. Click **Next**.
6. For **Scheduling**, you want **Now** (unless you choose another time). Then click **Next**.
7. Accept all the other default options and then click **Review** at the bottom of the screen.
8. Re-read your message to verify that all is OK, then click **Publish** to send it. The message will be sent to all the app users (if they are connected or connect to the internet).

26. User Accounts

You can configure your app so that a user's highlights, notes and bookmarks are saved to a database server. This means that the user can see them on another device, and they do not need to lose their work when they change their phone.

To enable this feature, you need to do some configuration work in two places:

- within Scripture App Builder, and
- in the Google Firebase console.

26.1. Setting up User Accounts in Scripture App Builder

In Scripture App Builder:

1. Go to the **Data & Analytics** ➤ **User Accounts** tab.
2. Click on **Enable user accounts**.

26.2. Setting up the database in the Google Firebase console

To set up the database, which will contain the registered users' information, you need to add Firebase to your app, create a database, set up authentication and configure rules.

Add Firebase to your app

Follow the instructions in setting up a Firebase project, described in the section above on [Firebase Analytics](#).

Create a Database

To create a database:

1. In your Firebase project console, select **Realtime Database** from the menu on the left of the screen (the option below Firestore Database).
2. Click the **Create database** button.
3. On the first page of **Set up database**, choose where you would like the data to be stored, then click **Next**.
4. On the second page of **Set up database**, choose **Start in locked mode**, and click **Enable**.

Set up Authentication

To set up authentication:

1. In your Firebase project console, select **Authentication** from the menu on the left of the screen.
2. Select the **Sign-in method** tab.

3. Click on **Email/Password** and enable this authentication method to allow users to sign up using their email address and password.
4. Click **Save** to confirm your changes and return to the Sign-in method tab.

Configure Rules

Rules are required to tell Firebase who will have access rights to read and write to the database. To configure the database rules:

1. In your Firebase project console, select **Realtime Database** from the menu on the left of the screen.
2. Select the **Rules** tab.
3. Change the rules to be as follows:

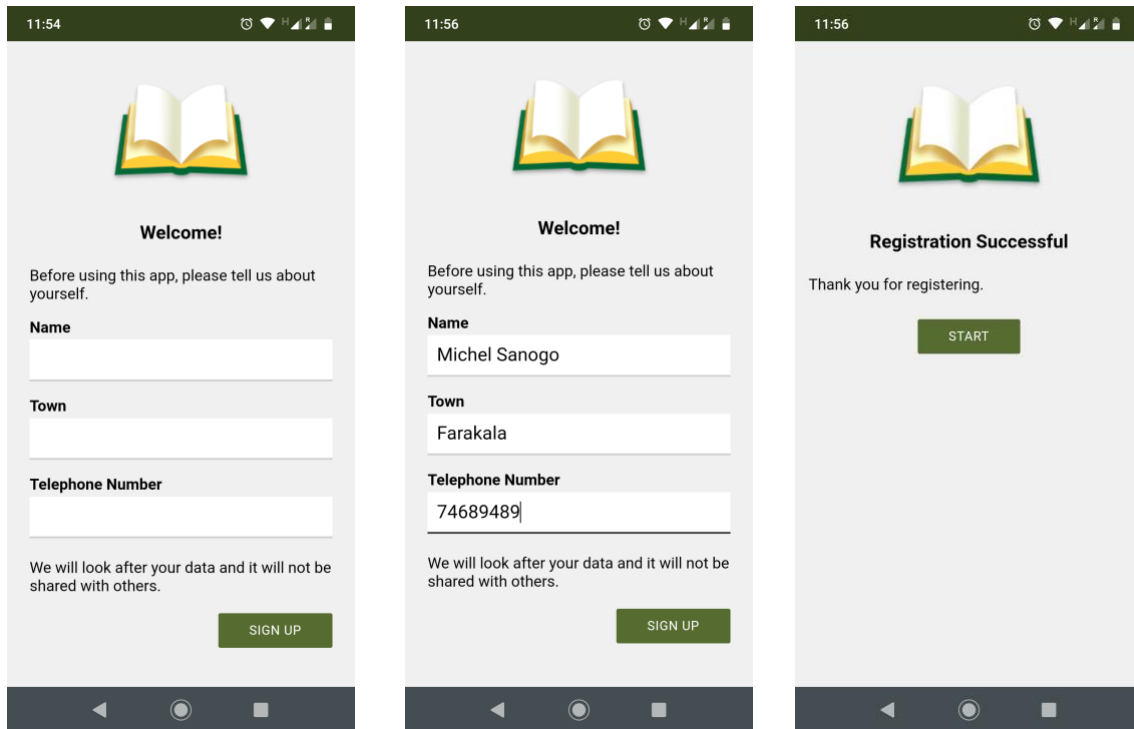
```
{
  "rules": {
    "authenticated-users": {
      "$uid": {
        ".read": "$uid === auth.uid",
        ".write": "$uid === auth.uid"
      }
    }
  }
}
```

This rule says that only authenticated users will have right to read and write their own data. Unauthenticated users will not be able to see or change anything, and authenticated users will not have the right to see or change someone else's data.

4. Click **Publish** to confirm your changes.

27. Registration Screen

You can define a registration screen to be shown when a user launches the app for the first time. This allows you to collect contact information, for example to connect people with Bible reading/listening discussion groups.



To set up a registration screen, you need to do some configuration work in two places:

- within Scripture App Builder, and
- in the Google Firebase console.

27.1. Setting up the Registration Screen in Scripture App Builder

To set up the registration screen within the app builder:

1. Go to the **App** ➤ **Security** page.
2. Select **Require each user to register with their details when they first use the app**.
3. Click the **Configure Registration** button.
4. Follow the instructions in each of the tabs in the Configure Registration dialog:
 - **Registration Screen:** specify the title and text to show on the registration screen.
 - **User Details:** specify the input fields (such as name, telephone number, email address), that you will ask the user to provide. You can specify which of these are required and which are optional.

- **Skip Registration:** choose whether you want the user to be able to skip the registration process. If they do press the Skip button, specify when you want the app to ask them again.
- **Registration Completed:** specify the title and text to be displayed on the screen which is shown to the user after they have successfully registered.
- **Image:** specify an image to be displayed at the top of the screen, such as your organisation's logo or the app icon.
- **Database:** choose whether the user data will be stored in an app-specific path in the database (which is useful if you use the same database for several apps), or at the top level of the database (which is fine if you have just one app). More information about configuring your database can be found in the following section.
- **Styles:** you can adjust the styling of the screens by adjusting the style declarations.

Use the **Test** buttons on the **Registration Screen** and **Registration Completed** tabs to preview the screens in a browser.

5. Click **OK** when you are finished in the dialog.

27.2. Setting up the database in the Google Firebase console

To set up the database, which will contain the registered users' information, you need to add Firebase to your app, create a database, set up authentication and configure rules.

Add Firebase to your app

Follow the instructions in setting up a Firebase project, described in the section above on [Firebase Analytics](#).

Create a Database

To create a database:

1. In your Firebase project console, select **Realtime Database** from the menu on the left of the screen (the option below Firestore Database).
2. Click the **Create database** button.
3. On the first page of **Set up database**, choose where you would like the data to be stored, then click **Next**.
4. On the second page of **Set up database**, choose **Start in locked mode**, and click **Enable**.

Set up Authentication

To set up authentication:

1. In your Firebase project console, select **Authentication** from the menu on the left of the screen.
2. Select the **Sign-in method** tab.
3. Click on **Email/Password** and enable this authentication method. This is used when viewing registered users in a web browser
4. Click **Save** to confirm your changes and return to the Sign-in method tab.
5. Click on **Anonymous** (at the bottom of the Sign-in providers list) and enable this authentication method. This is used during user registration in the app.
6. Click **Save** to confirm your changes and return to the Sign-in method tab.

Configure Rules

Rules are required to tell Firebase who will have access rights to read and write to the database. To configure the database rules:

1. In your Firebase project console, select **Realtime Database** from the menu on the left of the screen.
2. Select the **Rules** tab.
3. The rules you specify will depend on the path you have chosen on the **Database** tab in the Registration configuration.

If you have chosen to store the user information in an **app-specific location**, e.g. `"/apps/your-app-package-name/users/"`, the rules need to be as follows:

```
{
  "rules": {
    "apps": {
      "$package": {
        "users": {
          "$uid": {
            ".read": false,
            ".write": "$uid === auth.uid"
          }
        }
      }
    }
  }
}
```

If you have chosen to store the user information at the **top level of the database**, e.g. `"/users/"`, the rules need to be as follows:

```
{
  "rules": {
    "users": {
      "$uid": {
        ".read": false,
        ".write": "$uid === auth.uid"
      }
    }
  }
}
```

These rules give write access only to the user who is making the registration. No one will have read access (except for you when you view the database from the Firebase console).

Tip: Make sure you copy the rules exactly. An easy way to get the rules you need is to click the **Database Rules** button on the Database tab. This will give you the rules you can copy.

Note: If you are also using **User Accounts** in this app (see the previous section), you will need to combine the database rules for the registered users (Registration screen) with the rules for the signed up users (User Accounts). Please see below, **Rules FAQ**, for more details.

4. Click **Publish** to confirm your changes.

Rules FAQ

Q. If we have also enabled User Accounts for this app, what rules do we need to use?

If you are using both *User Accounts* and a *Registration Screen*, you need to combine the rules as follows:

```
{
  "rules": {
    "apps": {
      "$package": {
        "users": {
          "$uid": {
            ".read": false,
            ".write": "$uid === auth.uid"
          }
        }
      }
    },
    "authenticated-users": {
      "$uid": {
        ".read": "$uid == auth.uid",
        ".write": "$uid == auth.uid"
      }
    }
  }
}
```

The above rules assume that you want to store the user information in an app-specific location. If you have chosen the top-level location, you need to remove the “apps” and “\$package” elements above.

Q. If we want an administrator to have read access to the Registered users to display in a web browser, what rule do we use?

If you want to give an administrator read access to the data, to be displayed in a web browser (See **Configure Registration** ➤ **User Details** ➤ **View Data**), here is the rule to use:

```
{
  "rules": {
    "apps": {
      "$package": {
        "users": {
          "$uid": {
            ".read": "(auth != null) && (auth.token.email == 'me@abcd.com')",
            ".write": "$uid === auth.uid"
          }
        }
      }
    }
  }
}
```

(where me@abcd.com is a user added on the **Authentication** page)

The above rules assume that you want to store the user information in an app-specific location. If you have chosen the top-level location, you need to remove the “apps” and “\$package” elements above.

28. EPUB

As well as building a smartphone app, you can create e-books in EPUB format. EPUB documents are readable by a number of e-book readers on Windows, Mac, iOS and Android platforms.

If you have timing files for your audio, the e-books can contain ‘Read Aloud’ text-audio synchronization. These are readable by a few EPUB 3 readers, such as [Adobe Digital Editions](#) or [Thorium Reader](#).

To create an EPUB e-book:

1. Choose whether to create a single e-book containing all the books or whether you want to create a separate e-book for each book. The latter is the best option if you will have audio in the book, since otherwise a whole NT with audio would be a very large e-book.
2. Choose a cover design. See the **Cover** tab. If you are creating separate e-books for each book, a different cover image can be specified for each book on the Cover tab of each individual Book page.
3. Go to **Publishing** ➤ **E-books (EPUB)** and click the button **Create EPUB Document(s)**.

The generated EPUB documents will be saved in the default EPUB output folder. See **Tools** ➤ **Settings...** ➤ **Default Folders**.

4. Open the EPUB documents in your chosen e-book reader.

Scripture App Builder generates EPUB 3 documents which should be backwardly compatible to be able to be read using EPUB 2 readers. EPUB 3-only features such as the text-audio synchronization will only be seen on compatible readers.

"ᐃᐃᐃ yilᐃᐃ Lago 'na 'Denaade
Galile -sele- -ka me. ¹⁵ᐃ 'neᐃ:
«Zle -yle -ke ᐃᐃᐃᐃ "bhuᐃ, "ne -
Lago 'na Kamale Zikpe "ne -ka
la 'anye 'detlole-; 'zaa, 'a
bhelia- 'anye 'kose -nyenye -we
'ᐃ, -saa 'kele- Lago 'na
'Denaade 'dle "ku.»

**Zezu 'la 'ᐃᐃᐃ gelenya 'na
tetenoa**

(Mat 4.18-22; Luk 5.1-3,10-11)

¹⁶ Zezu ᐃᐃᐃᐃ- Galile 'na -jinye

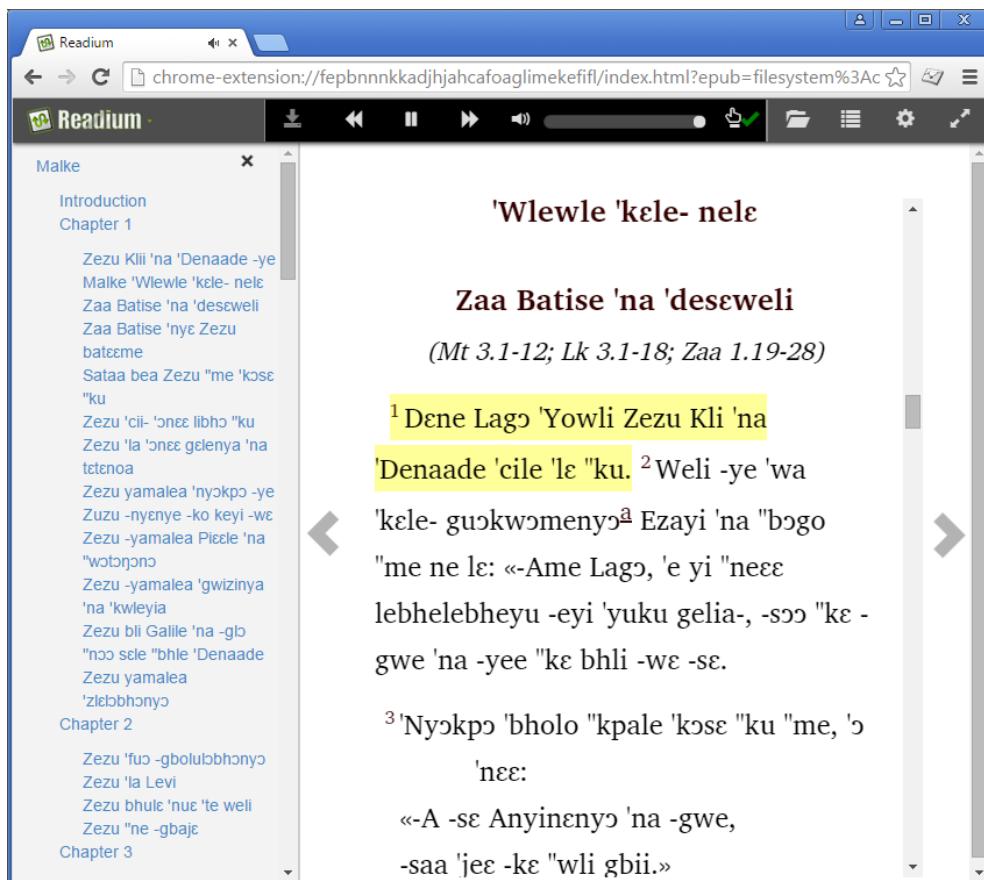
"ᐃᐃᐃ yilᐃᐃ Lago 'na 'Denaade
Galile -sele- -ka me. ¹⁵ᐃ 'neᐃ:
«Zle -yle -ke ᐃᐃᐃᐃ "bhuᐃ, "ne -
Lago 'na Kamale Zikpe "ne -ka
la 'anye 'detlole-; 'zaa, 'a
bhelia- 'anye 'kose -nyenye -we
'ᐃ, -saa 'kele- Lago 'na
'Denaade 'dle "ku.»

**Zezu 'la 'ᐃᐃᐃ gelenya 'na
tetenoa**

(Mat 4.18-22; Luk 5.1-3,10-11)

¹⁶ Zezu ᐃᐃᐃᐃ- Galile 'na -jinye

Viewing an EPUB Read Aloud document in the Gitden Reader app



Viewing an EPUB Read Aloud document in the Radium Chrome extension

29. Progressive Web App

In addition to building a smartphone app, you can create two separate types of [Progressive Web App](#) (PWA). The first type (Static PWA) uses the “Generate HTML” static HTML output of Scripture App Builder originally used to test text/audio synchronization. The second type (Modern PWA) is a modern web application that replicates many of the smartphone app features.

For a PWA to be installable, it must be published to a web server that serves the files using HTTPS.

29.1. Why Would I Want a Progressive Web App?

Instead of being distributed via the Google Play Store or Apple App Store, PWAs are published as websites. They can be installed from the browser on all platforms, including mobile (iOS and Android) as well as desktop (Windows, Mac, Linux).

29.2. Which Progressive Web App Should I Use?

The Static PWA uses the Java code from the smartphone app to generate the HTML from the USFM/USX files so that the HTML matches closely the display in the smartphone app. However, it is limited to a single book collection and audio location, the book/chapter/verse navigation is simple, and there are minimal app features implemented. When used as a PWA, it can take a long time for all of the static HTML files to be downloaded by the browser. It may take quite a while before it can be added to the home screen of the device and used offline.

The Modern PWA is a proper web app implementation with support for multiple book collections, all the audio sources (including streaming from FCBH BibleBrain), and many of the features of the smartphone app. It is not complete yet but will be improved over time. It uses a separate JavaScript library to read the USFM/USX files and display the content as HTML. There may be differences in the HTML output that will be improved as well. Due to how the Modern PWA is designed, the download will be significantly faster and be available quickly to be added to the home screen of the device and used offline.

29.3. Audio in PWA

If your project uses WebM (Ogg Opus) audio format for some or all of the audio files, this format is not supported by Safari on iOS and some older versions of macOS (it works with Safari 16.0 or later). Both PWAs will fall back to MP3 and CAF audio formats if WebM is not supported. If you want the PWA to work with Safari on macOS or iOS, make sure to include an MP3 or CAF file for each WebM file with the same base filename (e.g. mat1.mp3 or mat1.caf along with mat1.webm). Remember, all browsers on iOS use the Safari web browser component internally so this affects all browsers on iOS.

When using an FCBH Bible Brain audio source with the Modern PWA, if you use a Fileset ID for WebM audio, it will have “-opus16” at the end (e.g. ENGWEBN2DA-opus16). Since this will not work on iOS, the Modern PWA will attempt to use the MP3 equivalent.

29.4. Building a Static PWA

29.4.1. Installing Prerequisites for Static PWA on Windows

In order to create the files needed for a PWA, you need to have 2 components installed on your computer:

1. **Node.js**
2. **Workbox-cli**

Here are more details on installing both of these components.

Go to the Node.js Downloads page:




<https://nodejs.org/en/download/>

There are many download files on this page. You are looking for the file that corresponds to your computer’s operating system type. It is easiest to download the msi (installer package file) rather than the zip file.

Downloads

Latest LTS Version: **12.17.0** (includes npm 6.14.4)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features	
 Windows Installer <small>node-v12.17.0-x86.msi</small>	 macOS Installer <small>node-v12.17.0.pkg</small>	 Source Code <small>node-v12.17.0.tar.gz</small>

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8

When the file has downloaded, run it to install Node.js on your computer.

After installing Node.js, bring up a command prompt. Click on the Windows (start) icon and type `command prompt` and press Enter. Paste in this command and press enter:

```
npm install workbox-cli -global
```


Once this command finishes running, you have the prerequisite tools for exporting PWA files. Close SAB if it is open and restart it.

Sometimes, there may be issues installing `workbox-cli`. If this happens, you will be unable to check the **Include Progressive Web App files** checkbox mentioned below. To fix this issue clear the npm cache by running the following command:

```
npm cache clean -force
```

And then re-run the above install command.

29.4.2. Installing Prerequisites for Static PWA on Linux

In order to create the files needed for a PWA, you need to have 2 components installed on your computer:

1. **Node.js**
2. **Workbox-cli**

Here are more details on installing both of these components.

First, find the most recent Long Term Stable (LTS) release version of Node.js. Go to:

<https://nodejs.org/en/download/>

And find the line near the top of the page that begins “Latest LTS Version:”.



Take note of the number before the first dot, as this is the major version you want to install.

Follow the instructions at:

<https://github.com/nodesource/distributions>

to install that major version for your Linux distribution.

Once you have Node.js installed, open a Terminal window and run the following command:

```
sudo npm install workbox-cli -global
```

Once this command finishes running, you have the prerequisite tools for exporting PWA files. Close SAB if it is open and restart it.

Sometimes, there may be issues installing workbox-cli. If this happens, you will be unable to check the **Include Progressive Web App files** checkbox mentioned below. To fix this issue clear the npm cache by running the following command:

```
npm cache clean -force
```

And then re-run the above install command.

29.4.3. How do I export Static PWA files?

To export PWA files, you have to first have to configure the PWA settings. In SAB, right click the collection that you want to export and select **Export to HTML Settings**. In the resulting window, select the **Progressive Web App** tab.

In this tab, check the **Include Progressive Web App files** checkbox. If it is grayed out, ensure you have installed Node.js and Workbox-cli, then see the above note about clearing the npm cache.

Enter a **Name** and **Short Name**. Click **OK**.

Right click the collection again and select **Export to HTML**.

The generated PWA files will be saved in the default HTML output folder. See **Tools > Settings... > Default Folders**.

29.4.4. How do I Add Audio to my Static PWA?

Since PWAs are installed to devices through the internet, they cannot rely on audio files stored locally on your machine. You will need to host audio files through an HTTPS server so that the PWA can download it from any device. HTTPS is preferred over HTTP since some devices will reject HTTP downloads.

If you want your PWA to have audio capabilities, right click on the collection you want to export as a PWA. Select **Export to HTML Settings**. In the resulting window, you should be on the **Audio Files** tab.

Select the **Use the following path to the files, e.g. a URL address** menu button and enter the URL from which your audio files can be downloaded.

29.5. Building a Modern PWA

Scripture App Builder includes all the software to build a Modern PWA. There are no additional prerequisites.

29.5.1. Configuration for Modern PWA

A Modern PWA uses the existing configuration of the smartphone app. There is an additional requirement to specify the Language for the Book Collections. Go to the **Books > Book Collection Name > Language** page and specify the **Language Code**.

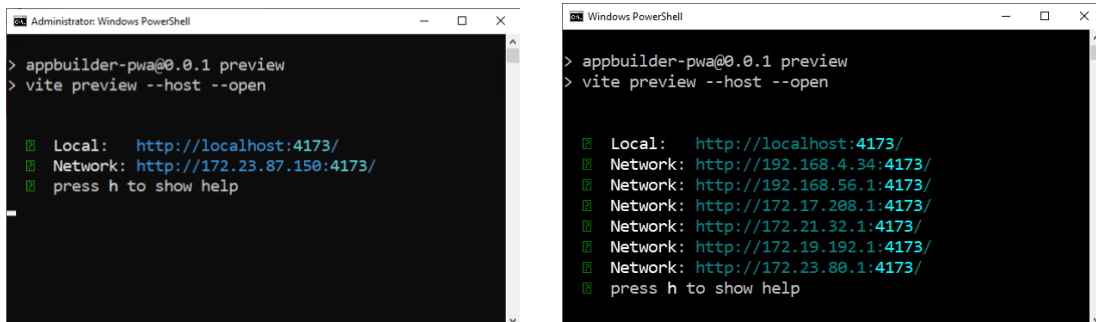
If the PWA will be published to a subdirectory of a web server (e.g. <https://example.com/bible> instead of <https://example.com>), go to the **Publishing** ➤ **Web App (PWA)** ➤ **Subdirectory** page and specify the subdirectory (e.g. /bible).

29.5.2. How Do I Build a Modern PWA?

Building a Modern PWA is like building a smartphone app. Click the **Build PWA App** button on the toolbar at the top of the screen. If something is not configured correctly for the build to work, you will be notified. A compilation window will appear. Wait while the PWA is compiled. When it completes successfully, go to **File** ➤ **View PWA Output Folder** to find the output files from the PWA build.

29.5.3. How Do I Test a Modern PWA?

With the Static PWA, you can open the Static HTML files from your computer. With the Modern PWA, you need a web server to run it. To preview the Modern PWA, click on the Run PWA App button on the toolbar at the top of the screen. A compilation window will appear. When the build is complete, a terminal window and a browser window will appear. The terminal will run the PWA on a local web server. The terminal window will display the URLs of the local web server. If your smartphone is on the same network as your computer, you can test the PWA on your smartphone by opening a web browser on your smartphone and entering the URL for the entry labeled **Network** (e.g. <http://172.23.80.88:4173> in the screenshot on the left). If there are multiple **Network** entries (e.g. Windows running Windows Subsystem for Linux on the right), then start trying them from the top of the list (e.g. <http://192.168.4.34:4173>).



```
> appbuilder-pwa@0.0.1 preview
> vite preview --host --open

Local: http://localhost:4173/
Network: http://172.23.87.150:4173/
press h to show help
```

```
> appbuilder-pwa@0.0.1 preview
> vite preview --host --open

Local: http://localhost:4173/
Network: http://192.168.4.34:4173/
Network: http://192.168.56.1:4173/
Network: http://172.17.208.1:4173/
Network: http://172.21.32.1:4173/
Network: http://172.19.192.1:4173/
Network: http://172.23.80.1:4173/
press h to show help
```

Make sure to close the terminal window when you are done testing the PWA.

29.5.4. How Do I Publish a Modern PWA?

To publish a Modern PWA, copy the files from the **PWA Output Folder** to a web server connected to the Internet. For the PWA to be installable on a smartphone or desktop computer, the web server needs to serve the files using HTTPS. PWA installation works with supporting browsers. See <https://web.dev/learn/pwa/installation/> for more details.

30. Publishing and Distribution

Before you distribute your app, please use the *App Publishing Checklist* (available via the Help menu in SAB or on the SAB website) to verify that it is ready to publish.

Apps built with Scripture App Builder can be published on the Google Play store, distributed by memory card, shared by Bluetooth or Wi-Fi transfer, uploaded to websites, or sent out by email. For more information, please see the user manual *Distributing Apps* (available via the Help menu in SAB or on the SAB website).