

**http://localhost:3000**

# Table of contents:

- How to Use This Manual
- Preparing Content for your app
- Preparing Text
- Paratext files for Bible books
- Other SFM text files
- Word Documents (.docx)
- Digital Bible Library Text Release Bundles
- HTML files (.html)
- Bloom books (.bloompub)
- Preparing images
- Preparing audio
- How to build your first app
- Installing the app on your phone
- What sort of apps can I build?
- Scripture (Bible) apps
- Picture Storybook apps
- Lectionaries
- A collection of Bloom books
- Song Book apps
- Quiz apps
- Hybrid apps (combining different app modules)
- How can I customize my app to make it contextualized for a specific audience?
- How can I find the setting or feature that I need in my SAB project?
- How should I choose the app package name?
- Why and how should I add multiple translations to my app?
- Where do the default book names and abbreviations come from?
- Do I have to create a new keystore for each app, or can I reuse the same keystore for several of my apps?
- How can I include pictures in my apps?
- How can I handle a “right-to-left” language?
- Setting the UI layout direction
- Setting a book collection’s text direction
- Setting a specific book’s text direction
- Changing the Orientation of the Contents Menu
- Can I build apps when I do not have internet access?

- How do the glossary entry popups work?
- Can I build an app from the command line?
- How can I add a horizontal line to separate certain paragraphs in text?
- I don't like the name "Scripture App". Have you thought of calling the app something else?
- SAB Menu Bar
- New App
- Open App...
- Open Multiple Apps...
- Open App From Scriptoria
- Close
- Save
- Save All
- Copy App
- View Project Folder
- View APK Output Folder
- View IPA Output Folder
- View PWA Output Folder
- Exit
- Build Android App APK
- Build Android App Bundle
- Build Progressive Web App (PWA)
- Build iOS Asset Package
- Build Multiple Apps
- Install Latest APK on Device
- Export Data Safety to CSV
- Settings
  - The Code Libraries tab
  - The Android SDK tab
  - The Defaults tab
  - The Default Folders tab
  - The Export Folders tab
  - The Build Settings tab
  - The Build Folders tab
  - The After Build tab
  - The Interface tab
  - The Files tab

- The Slideshows tab
- Create New Keystore
- Install JDK
- Install Android SDK
- View Logs
- Check aeneas installation
- The Help menu
- Find Settings and Features
- Help Documents
- Community Website
- About App Builder
- The SAB Tool Bar
- New App...
- Save
- Save All
- Build Android App
- Install APK
- Build PWA
- Run PWA
- The App project menu
- App Name
- Package
- Project
- APK
- The Android Versions tab
- The Install Location tab
- The Multiple APKs tab
- iOS
- App Signing
- Expiry
- Security
- The About project menu
- What information should I include in the About page?
- Which formatting codes can I use in the About page?
- Which variables can I use in the About page?
- Icons
- The Android Icon (Adaptive) tab

- The Android Icon (Legacy) tab
- The Notification Icon tab
- Graphics
- The Android Splash Screen tab
- The Navigation Drawer tab
- The Borders tab
- Interface
- The Languages tab
- The Translations tab
- The Layout Direction tab
- Colors
- The Color Scheme tab
- The Main Colors tab
- The Book Colors tab
- Fonts
- The Main Font tab
- The Font Files tab
- The Font Handling tab
- The Keyboards tab
- The Lexical Models tab
- Styles
- The Text Styles tab
- The User Interface Styles tab
- The Custom Styles tab
- Setting styles at the book collection or book level
- The Formatting tab
- The Text Size tab
- Changes
- Analytics
- FCBH Digest Analytics
- S3 Digest Analytics
- Data Payload for FCBH and S3 Digest Analytics
- Amplitude Analytics
- Firebase Analytics
- Firebase
- The Firebase Features tab
- The Firebase Configuration tab

- The Push Notifications tab
- The In-App Messaging tab
- User Accounts
- Setting up User Accounts in Scripture App Builder
- Setting up the database in Google Firebase console
- The Data Deletion tab
- The Publishing project menu
- App Store
- Web App (PWA)
- E-books (EPUB)
- Contents Menu
- The Contents Items and Screens tab
- The Contents Settings tab
- The Viewer tab
- Navigation Drawer
- The Menu Items tab
- The Settings tab
- The Sharing tab (sharing apps)
- The History tab
- The Search tab
- Bottom Navigation
- Book Navigation
- The Navigation tab
- The First Book and Chapter tab
- The References tab
- Deep Linking
- Setting up Deep Linking
- Creating Deep Links
- Deferred Deep Linking
- Tab Types (Book Tabs)
- The Engagement project menu
- Annotations
- Assistant (AI)
- Notifications
- The Verse of the Day tab
- The Verses tab
- The Daily Reminder tab

- The Notification Icon tab
- Bible Reading Plans
- The Plans tab (for adding and modifying plans)
- The Plans Settings tab
- The Plans Viewer tab
- Verse on Image
- The Verse on Image tab
- The Backgrounds tab
- The Watermarks tab

# How to Use This Manual

Scripture App Builder (SAB) has many features to help users engage with the Scriptures. It is recommended that you at least familiarize yourself with the many options available by taking a brief look over all the chapter topics.

If you are new to SAB, it is recommended that you read sections 1, 2 and 3 of this manual before running the New App wizard for the first time. Section 4 has some Frequently Asked Questions. Sections 5 and on systematically cover the menus, project features and configuration settings.

This guide includes notes and tips to help you get the most out of the software. Tips and notes will appear like this:

## Notes:

### NOTE

*Each page in the wizard will have explanatory text to help you in choosing options.*

## Tips:

### TIP

*To center text in the About box, add “**text-align: center;**” to the CSS in the **body.about** UI style.*

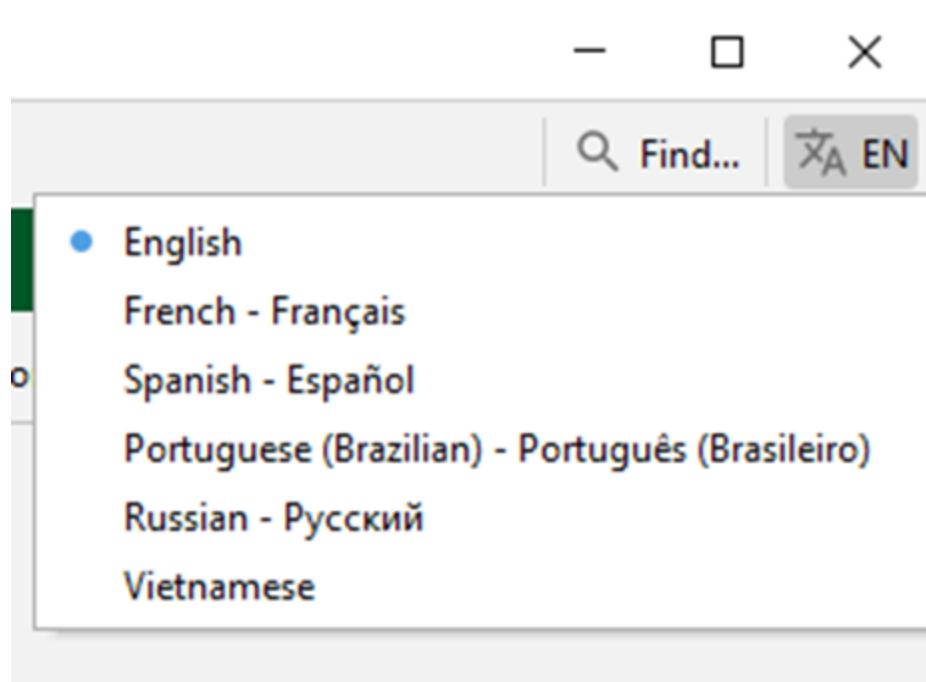
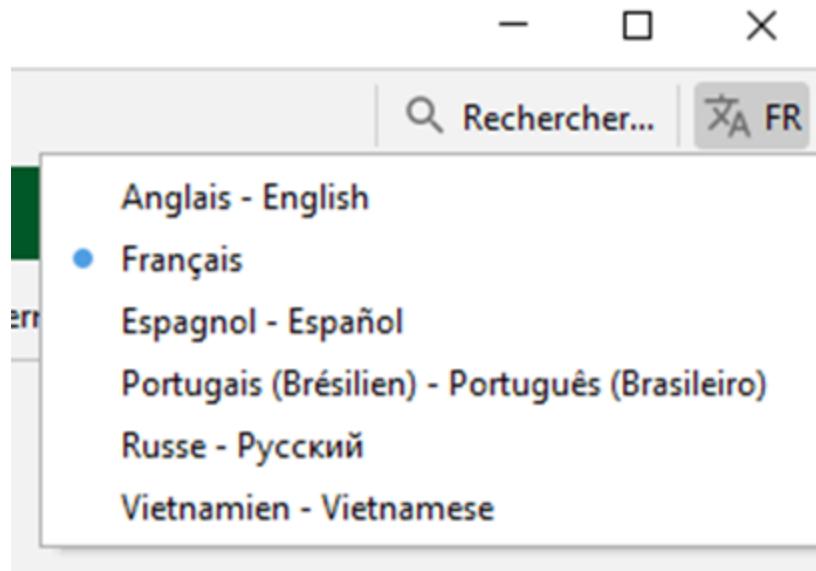
Most of the interface items in your app can be customized through the User Interface (UI) styles. For each section, the related UI styles will be shown in text boxes like the following:

To modify this...	Adjust this style...
Navigation drawer background	ui.drawer
Navigation drawer item text	ui.drawer.item.text

See Appendix 3 for a comprehensive list of styles and explications.

SAB includes a helpful search feature to aid in finding the various features and settings. Click on **Find...** at the right side of the SAB tool bar. See also Find Settings and Features...

To change SAB's interface language, click on the language button at the far right side of the SAB tool bar. See the example below:



The SAB software interface has four main areas: The Menu bar, the Tool bar, the Project menu area (tree view) and the Configuration area. These will be referenced in the various chapters.

The screenshot displays the Scripture App Builder application window. At the top, there is a menu bar with options: File, Build, Tools, Help, followed by a title bar labeled "Scripture App Builder". Below the title bar is a toolbar with buttons for New App..., Save, Save All, Build Android App, Install APK..., Build PWA, and Run PWA. On the far right of the toolbar is a language switcher showing "EN".

The main area is divided into two sections:

- Project Tree View (Left):** A tree view of the project structure. It starts with "Apps" and includes branches for "SAB Scripture Demo App", "Brazzaville", "beq-Beembe Bible", "Wamey", "Wamey Bible Songs", "CI Apps", and "sev-Nyarafolo Bible". The "sev-Nyarafolo Bible" node is selected, highlighted with a green border. Its children include "App", "About", "Appearance" (with sub-options: Icons, Graphics, Interface, Colors, Fonts, Styles, Formatting, Changes), "Data & Analytics", "Publishing", "Navigation", "Engagement", "Media" (with sub-options: Audio, Video, Illustrations, Radio), "Books", and "Niarafolo".
- Configuration Area (Right):** This area is titled "sev-Nyarafolo Bible: App". It contains tabs for App Name, Package, Project, APK, iOS, App Signing, Expiry, Security, and Permissions. The "App Name" tab is active, showing the current value "Niarafolo (Nouveau Testament+)". There is also a "Translations..." button. Below the tab bar, there is a section titled "Tips on choosing an app name:" with the following bullet points:
  - Keep the name short and simple, e.g. "Mamara Bible" rather than "The Bible in the Mamara language". Otherwise some of the name will be cut off when displayed on the phone.
  - Make sure the name is clear and understood by speakers of the language.
  - Use mixed case, e.g. "Mamara Bible" rather than all caps, "MAMARA BIBLE".
  - App names will be displayed using the system font on the device, so don't include characters that require a special font.

# 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.

# 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)

 **NOTE**

*For more information on how SAB stores your text files, see How SAB stores your project data files in Appendix 1*

Here are some more details about each of these formats:

# 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. See example below:

```
\c 5
\s Kilε Kafila yεrε lemu l'a pye faabobonč ki nujč ni ge
\p
\l 1 Ayiwa, ba Yesu ya sipyiire ti na wε, na dugi tiin faabobonč ki nujč ni. A wu wo
kalaapiire t'i fulo wu na.
\l 2 A wu nč kčn na pu kalaa na:
\s Piikε p'i nε duba nagoo pu wε?
\l (Luka 6:20-23)
\q1
\l 3 «Piimu p'a li cε na funmč fεε pee nε
\q2 Kilε shizhaa na ge,
\q2 pee nε duba nagoo,
\q1 bani pee wuuro ti nε
\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>.

For more information about how SAB and Paratext work together, see *Using SAB with Paratext files*.

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...**

# 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 in the form \id XYZ, where XYZ is a unique, alphanumeric code you choose (with no spaces). Do not choose a code already reserved for a Scripture book, e.g. GEN, or any other book in your project.

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 (which allows you to set the Encoding to UTF-8 when you save).

To create a text file on a Mac, useTextEdit, 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 23 and 24.

# 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.

 **NOTE**

*Even though Word documents may appear easier, they work best only for very simple story books. If you want more control over the formatting and features of the book, use the SFM format.*

For more details about how to define picture story books and song books using Word documents, please see the sections on *Picture Story Books* and *Song Books*.

# 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/>

# **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.

# 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>

# Preparing images

Images are imported automatically from Word documents or need to be specified separately if you use SFM format. They should be in JPEG or PNG format. Certain interface items require a particular format, size or shape, and that will be specified in the descriptive text on that screen in SAB. For example, the navigation drawer image is required to be a PNG file with an aspect ratio of 16:9.

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.

# 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.



*Make sure that audio files are clearly and uniquely named. SAB will need to figure out which audio file to associate with which page or chapter, so clearly include the number in the file name (i.e. at the end of the name) using a leading zero. E.g.*

***my\_book\_page\_01.mp3***

*Audio from Faith Comes By Hearing will already have unique file names, so it is not recommended to change them.*

# How to build your first app

To build your first app with Scripture App Builder, you should use the **New App** wizard.

## NOTE

*The wizard will offer many choices in setting up your app. If you do not yet know exactly what you need, just choose the default option for now. All the app elements can be added or modified later within SAB, before you publish the app. Each page in the wizard will have explanatory text to help you in choosing options.*

1. Launch **Scripture App Builder** from its icon on the desktop.
2. Click **New App** on the toolbar. The New App wizard will appear.

## NOTE

*If this is your very first app, or you have no projects opened in SAB, the **New App** button will show in the middle of screen.*

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 appear under the app icon on the user’s phone. Do not include underscores or hard to understand abbreviations and try to keep the length under about 20 characters. (You can modify this later under **App → App Name**.)

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. (You can modify this later under **App → Package**.)

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

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

5. On the **Book Type** page, choose if you want an app with text or an audio-only app. Most Scripture apps will have text. If you choose audio-only, the wizard will present several

screens asking which books you want to add and which image you want to display while the audio is playing. Then continue with step 7.

6. If you choose to add books with text, the next page of the wizard is 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. (You can add more books later under **Books** → [Collection Name] → Book Collection → Add Books.)

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

7. 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. (You can modify this later under **Books** → [Collection Name] → Book Order.)

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

8. If the book names are in all-caps, the next page of the wizard is titled **Book Names** and will allow you to change them to mixed case if desired.

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

9. 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. (You can modify this later under **Books** → [Collection Name] → Language.)

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

10. 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. 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 organization for advice. They will want to make sure you get this right and do not simply make a guess as to what to include. (You may add this information at a later time if you do not have it available right now under **Books** → [Collection Name] → Copyright & Licensing. Also see more on where and how to include copyright information in the *About Page* section.)

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

11. 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. (You can modify this later under **Appearance → Fonts → Main Font.**)

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

12. If this next page is titled **Choice of Fonts**, 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. (You can modify these settings later under **Appearance → Fonts → Font Files.**)

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

13. On the page of the wizard titled **Font Handling**, you can select GeckoView if you know that the standard Android components will have trouble displaying the text correctly (e.g. if it is a complex script). (You can modify this setting later under **Appearance → Fonts → Font Handling.**) More information on GeckoView can be found in the *Fonts* section of this document.

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

14. 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. (You can modify this later under **Appearance → Colors → Color Scheme.**) Individual colors for text, titles, links, backgrounds, etc. can be customised later.

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

15. On the page of the wizard titled **Default Interface Language**, choose the language you want users to see when they first enter the app (the language for the app menus). This can be the current system language, but you can specify a default language just in case--perhaps a *language of wider communication*. (You can modify this later under **Appearance → Interface → Languages.**)

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

16. On the page of the wizard titled **Interface Languages**, choose the app interface languages that the user can choose between in the app settings. (You can modify this later under **Appearance → Interface.**)

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

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

You can modify the details for the Verse of the Day (and the Daily Reminder) later under **Engagement → Notifications**.

You can modify the details for saving as a video later under **Engagement → Verse on Image**.

 **NOTE**

*If you have audio versions of the books in your app, you have the option of allowing users to share verse audio with others. They can also share a video that combines the text and audio together. See **Configuring Verse on Image in the App Builder** for more details.*

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

18. 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. (You can see more options for creating an icon and/or modifying the icon later under **Appearance → Icons**.)

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

19. On the page of the wizard titled **Signing**, you need to specify the keystore and alias to use to sign the app. A keystore is a digital certificate file that ensures the app is legitimate. An app must be signed in this way so that it can be installed on a phone.

If you already have created a keystore for another app, see *reusing a keystore*. 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.” Specify a password. Click **Next** to continue.

iii. Enter an alias name for a key to create within your new keystore, such as the default “key1”. Specify a password, which can be the same as the password you entered on the previous page.

**(i) NOTE**

*It is very important to securely store the password for your keystore. Once the app is published, it cannot be updated without the keystore password. Make sure someone else in your organization has access to the app keystore passwords.*

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**.

20. Back on the **Signing** page of the New App wizard, you need to specify the keystore password, select the key alias and enter the alias password--just as you entered them in the step above. (You can modify your keystore settings later under **App → App Signing**.)

Click **Next** to continue.

1. On the page of the wizard titled **Project**, you can enter or 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. (You can modify this later under **App → Project**.) See the *Project section* under the explication of the App menu.

**(i) NOTE**

*SAB will create a project folder using the title of your project. It is recommended that you don't include any special characters in the project name as they may not be compatible with the folder name rules.*

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

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.

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.

 **NOTE**

*If you need to see the error information at a later time (e.g. to send to one of the support staff for help), you can find it on the menu bar under **Tools** → **View Logs...** → **Error Log**.*

A compilation window will appear. Wait while the app is compiled. It may take several minutes or more.

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.

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

 **NOTE**

*To understand more about where SAB stores files like your APK file, see **How SAB stores your project data files**.*

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

# 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.

## NOTE

*Because you are installing your APK directly (a process called side-loading) instead of through the Play Store, your phone may alert you to the fact that this is a suspicious file and require an extra step for permitting the install and/or a virus scan. This is normal, but if you are sharing your APK this way, you will want to prepare users that they may encounter these checks.*

There are several ways to do this:

### **Copy the APK via WIFI transfer**

Use a WIFI transfer app (Xender, WhatsApp) on your phone and computer. Open the *APK output folder*, select your APK and copy it or send it to your phone. Locate the APK file and tap on it to install.

### **Copy the APK directly**

If you have a USB data cable, connect your phone to your computer so that it shows up as a device (like an external hard drive or USB drive). Open the *APK output folder*, select your APK and copy it to an easily-found folder (e.g. Downloads) on your phone. Then, on the phone, find the APK and tap to install.

### **Automatically install the APK from within SAB**

This is a very convenient way to test your APK and worth setting up. Use the following steps:

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.)

 **TIP**

*When you connect your phone to your computer, if your phone appears as an icon under "This PC", you have a data cable.*

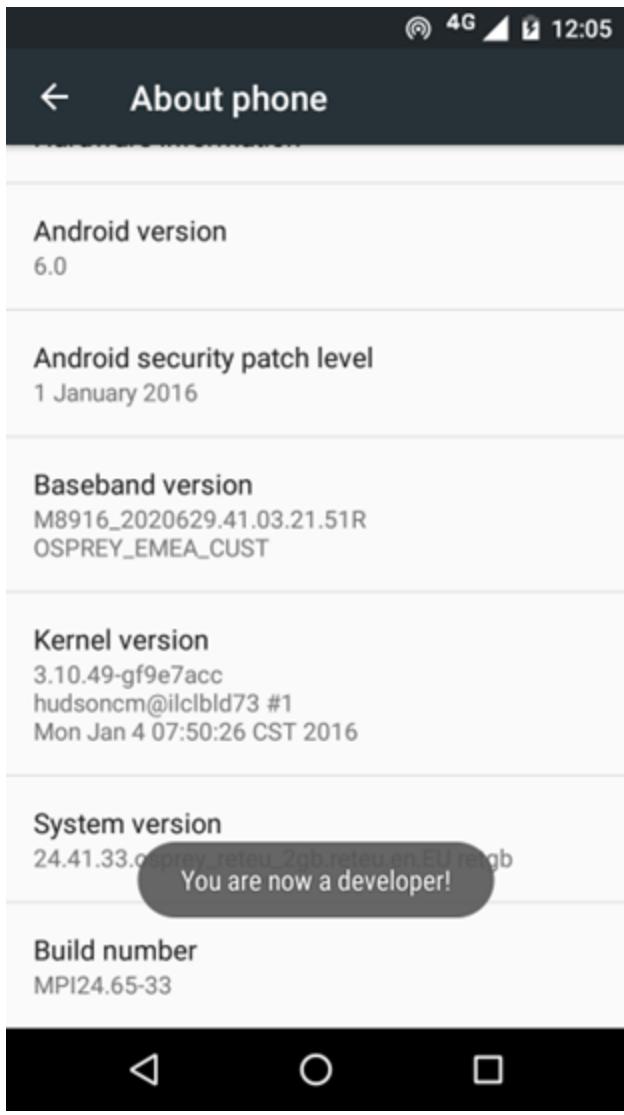
2. Ensure that **Developer Options → USB Debugging** is enabled on your phone. By default, on new phones, Developer Options is turned off. This is how you can enable it:

- 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'.

 **TIP**

*If you have trouble finding the Build Number, search for "**Build Number**" in the settings Search bar.*

- 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**.

#### TIP

*If you have trouble finding the Developer Options, search for “**Developer Options**” in the settings Search bar. You can even just search for “**USB Debugging**” to go directly to that setting.*

Different phones place Developer Options in different places, so look around your Configuration menu until you find it (see image below).

## Settings



Motorola ID

**System**

Date & time

Accessibility

Printing

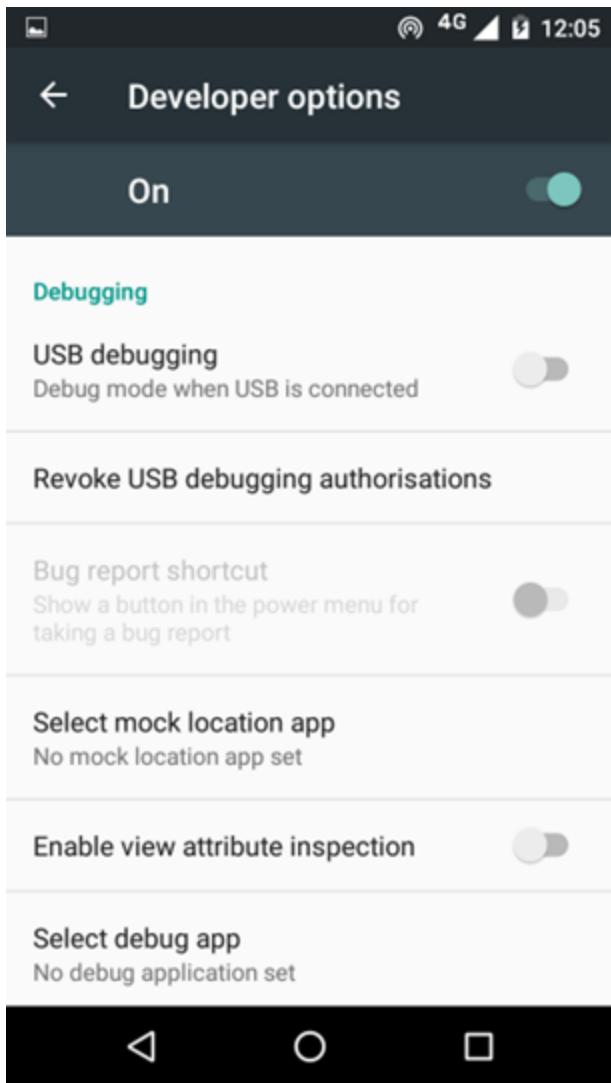
Developer options

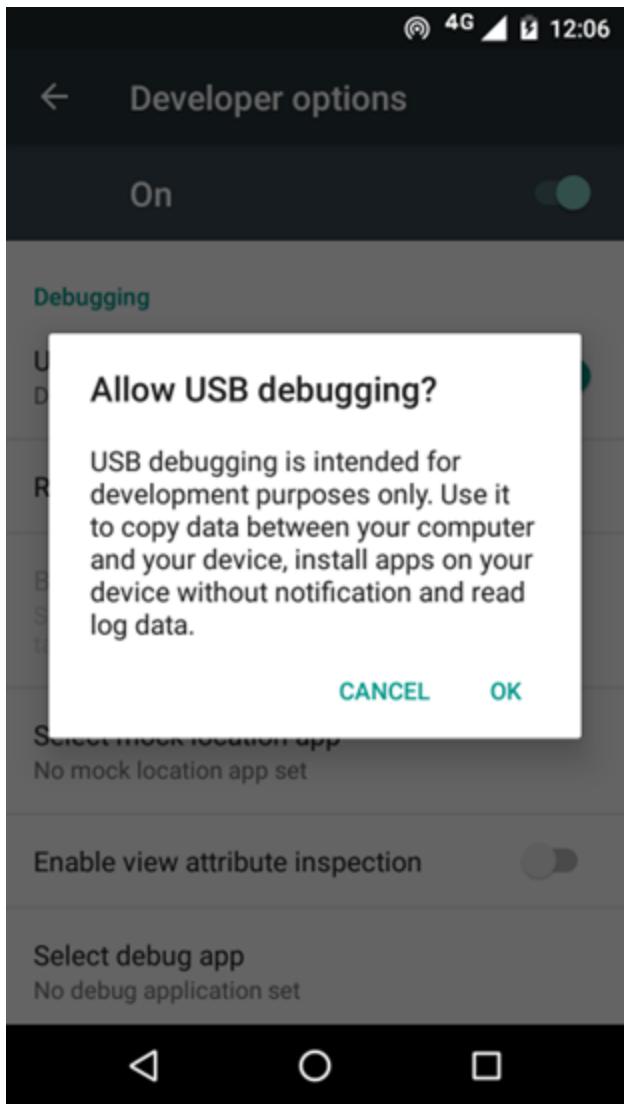
About phone



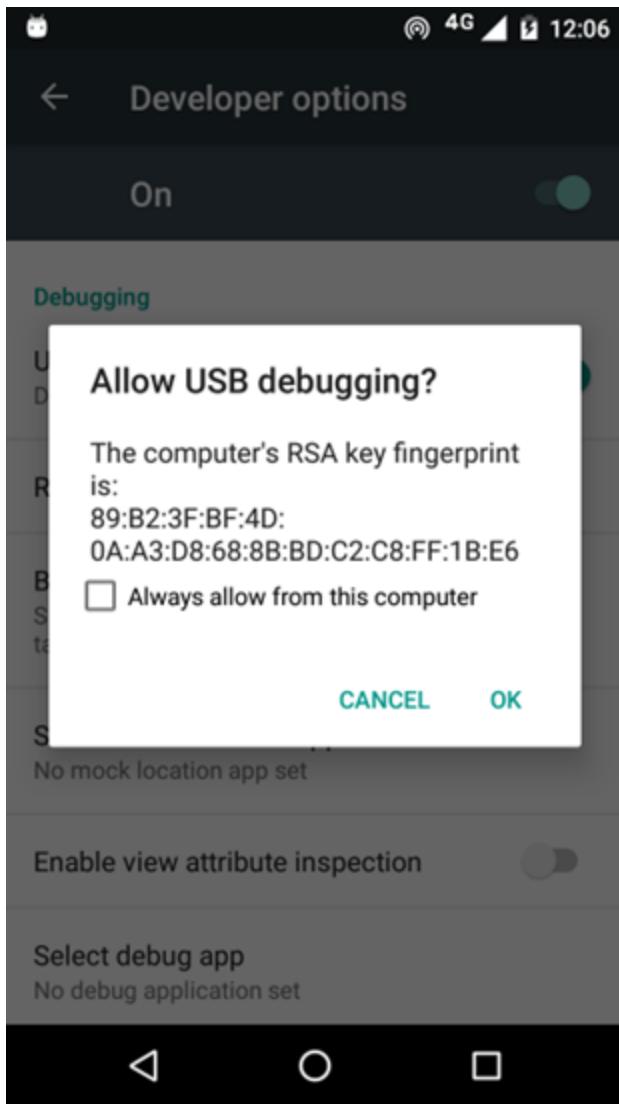
vi. Tap on **Developer Options** and ensure that it is turned on.

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, check the box “**Always allow from this computer**” and 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, your phone is locked, or that you have not enabled USB debugging on your phone.



*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.*

# What sort of apps can I build?

Scripture App Builder is very versatile, and you can build quite a few different types of apps including straight Scripture apps, Bible story books, lectionaries, literacy books song books and quiz books. You can even combine these different types of “modules” into a single app according to your wider [Scripture Engagement](#) strategy. Think of the “building blocks” that you have--Scripture texts, audio, videos, Bible study materials, Bible images—and prayerfully seek how to combine them in different ways in different apps to help people engage with God’s Word.

Use the links below to help you get more info on the various types of app types and modules you can build with Scripture App Builder:

# Scripture (Bible) apps

These are generally produced from Paratext files and are comprised of one or more Bible books with chapters and verses. See the following:

- [Problem Internal Link].
- [Problem Internal Link].
- [Problem Internal Link].
- Including [Problem Internal Link], [Problem Internal Link] and [Problem Internal Link].
- Providing [Highlighting, Notes and Bookmarks](#).
- Using [Problem Internal Link] for Bible study questions.
- Adding [Problem Internal Link].
- Adding an [Problem Internal Link].
- [Problem Internal Link].

# Picture Storybook apps

These comprise one or more storybooks, each generally with text and an image on each page. They are very versatile and can be used for illustrated Bible stories, literacy materials, Bible studies, Bible aids and any other illustrated resources. Looking at sites like [FreeBibleImages.org](http://FreeBibleImages.org) can provide some ideas and inspiration for books you can create.

To create storybooks, see the following:

- **[Problem Internal Link].**
- Preparing text for your storybook in [SFM](#) or [Word](#) formats, or [importing Bloom books](#).
- [Preparing your images](#).
- Using a **[Problem Internal Link]** to access your books
- **[Problem Internal Link].**
- Including **[Problem Internal Link]** for your text, images and other resources.
- **[Problem Internal Link].**

# Lectionaries

To create a 3-year lectionary, for example, make one book per collection (season) for each year, i.e. Advent would have 3 separate books, one for each year. These books would be in SFM format and can have the Bible verses typed out in full. However, if you add the lectionary books to a Bible app that has all the referenced Scripture books, you can put links to the actual Scripture texts in the lectionary SFM files.

See the following:

- Preparing text for your lectionary in [SFM](#) or [Word](#)
- Using a **[Problem Internal Link]** to access the books in your lectionary.
- Including **[Problem Internal Link]**.
- **[Problem Internal Link].**

# A collection of Bloom books

While the Bloom reader app gives access to any published Bloom books, it might fit your context better to have a contextualized app for a language group that includes just the Bloom books you want to group together. To do this, create an app with multiple books (in a single book collection) that you import from Bloom. Add a contents menu to clearly show which books are available.

See the following:

- [Importing Bloom books](#) into your app
- Using a **[Problem Internal Link]** to access the books.
- Including **[Problem Internal Link]**.

# Song Book apps

You can easily create a song / hymnbook app that has an automatically-generated menu allowing the user to select by song number or name. You can even put several songbooks in a single app if the songs are grouped in logical ways i.e. an Old Testament song book and a New Testament one.

See the following:

- [Problem Internal Link]
- Including [Problem Internal Link].
- Creating a [Problem Internal Link] to access several different song books.

# Quiz apps

For information on how to include quizzes in your app, see the document ***Creating Applications with Quizzes*** under the SAB **Help** menu.

# Hybrid apps (combining different app modules)

Depending on your Scripture Engagement goals, you may decide to create apps that have only one module (i.e. a songbook app) or have several modules (e.g. Scripture text with a lectionary, or a picture story book with several quizzes).

It is normally advised to put these different resources all under the same book collection if they are all in the same language. If you put them in separate collections, you will probably have to disable two-pane and verse-by-verse views under **Books → Layouts** as those are designed for parallel (linked) Scripture translations.

If you do have different types of resources in the same book collection, however, they will by default appear in the drop-down book selector (on the app menu bar) with all other books in the collection. This may be confusing for the user and/or result in a long and unwieldy list of books to scroll through. In this case, it is preferable to hide certain books from the list.

For example, if your app has a New Testament, a song book and several quizzes, it is best to have the Scripture books available in the drop-down book list and hide the others. You will then need to make a **[Problem Internal Link]** with links to the hidden resources so the user can access them.

To disable a book from appearing in the drop-down book selector, select the book in the project tree view, click on the **[Problem Internal Link]** tab and choose **No** under **Show in book selector**.

# How can I customize my app to make it contextualized for a specific audience?

One of the main advantages of SAB is the ability to completely customize each app for a specific audience. The following are links to the various app components that you may consider customizing:

- The **[Problem Internal Link]**
- **[Problem Internal Link]** including the splash screen, border and navigation drawer image
- Available **[Problem Internal Link]** and text direction
- **[Problem Internal Link]**
- **[Problem Internal Link]**
- Text and interface **[Problem Internal Link]**
- A **[Problem Internal Link]**
- Items available under the **[Problem Internal Link]**
- The **[Problem Internal Link]**
- What **[Problem Internal Link]** is available, if any
- Whether text should be **[Problem Internal Link]**
- What **[Problem Internal Link]** features are included
- Whether to include **[Problem Internal Link]** or **[Problem Internal Link]** in the text
- Whether to include **[Problem Internal Link]**
- Whether to include other resources like **[Problem Internal Link]**, quizzes, glossaries or other Bible aids.

# How can I find the setting or feature that I need in my SAB project?

SAB includes a helpful search feature to aid in finding the various features and settings. Click on **Find...** at the right side of the SAB tool bar. You can also search in this Building Apps document, which covers almost all of the available settings.

# How should I choose the app package name?

The app package name needs to be unique and cannot change once the app is published. For details on how to select a good package name for your app, see the *Package* section under the App Project Menu.

# Why and how should I add multiple translations to my app?

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 neighboring 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.

To add another translation, you will need to *create a new book collection*.

# 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):

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:

\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.



Reference Format

Book Names

Copy Book Names...

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
2CH - 2 Chronicles	2 Chr	2 Chronicles	2 Chronicles

Cross References (\xt) use

Abbreviation (toc3)

Parallel Passage References (\nr) use

Short Name (toc2)

On the Reference Format tab, specify punctuation used in Scripture references. On the Book Names tab, specify book abbreviations and book names used in Scripture references.

[More help...](#)[Show Guide](#)

OK

Cancel

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 found here:

**Books → [Collection Name] → [Book Name] → Book.**

# **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>

# 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**
2. Click Add Image and select the image file(s) to add. Then do one of the following:
3. Either 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, OR specify a reference to the image in a book file, using either the **\img** marker (which places the images at the top of the page and lets the text scroll independently underneath) or the **\fig...\fig\*** marker which puts the image in the text at the specified place (and it will scroll with the text). See *Picture Story Books* for more info on adding images in SFM (text) files.

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

# How can I handle a “right-to-left” language?

With Scripture App Builder it is possible to format your app all or in part as “left to right” (the default) or “right-to-left”, which is the norm for a number of languages including Arabic, Hebrew and Urdu.

# Setting the UI layout direction

The UI layout direction affects the position of the book name in the menu, and the side from which the navigation drawer slides out. To set the UI layout direction, See **Appearance → Interface → Layout Direction**.

# Setting a book collection's text direction

To set the text direction for an entire book collection, go to **Books** → **[Collection Name]** → **Styles** and select the direction from the **Text Direction** selector. This will not affect the UI, just the book texts.

# Setting a specific book's text direction

Occasionally, you may want to change the text direction on a book-by-book basis. To do this, click on the book name in the project tree view and select the **Styles** tab. Select the direction from the **Text Direction** selector.

# Changing the Orientation of the Contents Menu

You can switch the images in the contents menu to the right side of the menu screen and right-justify the text by changing the following styles:

To modify this...	Adjust this style...
The menu item's text background, style and justification (set justification to "right")	div.contents-text-block
The location of the image block. Set "float" to "right" to have the image on the right.	div.contents-image-block
The main Contents Menu. Set direction to "RTL."	body.contents

# 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:

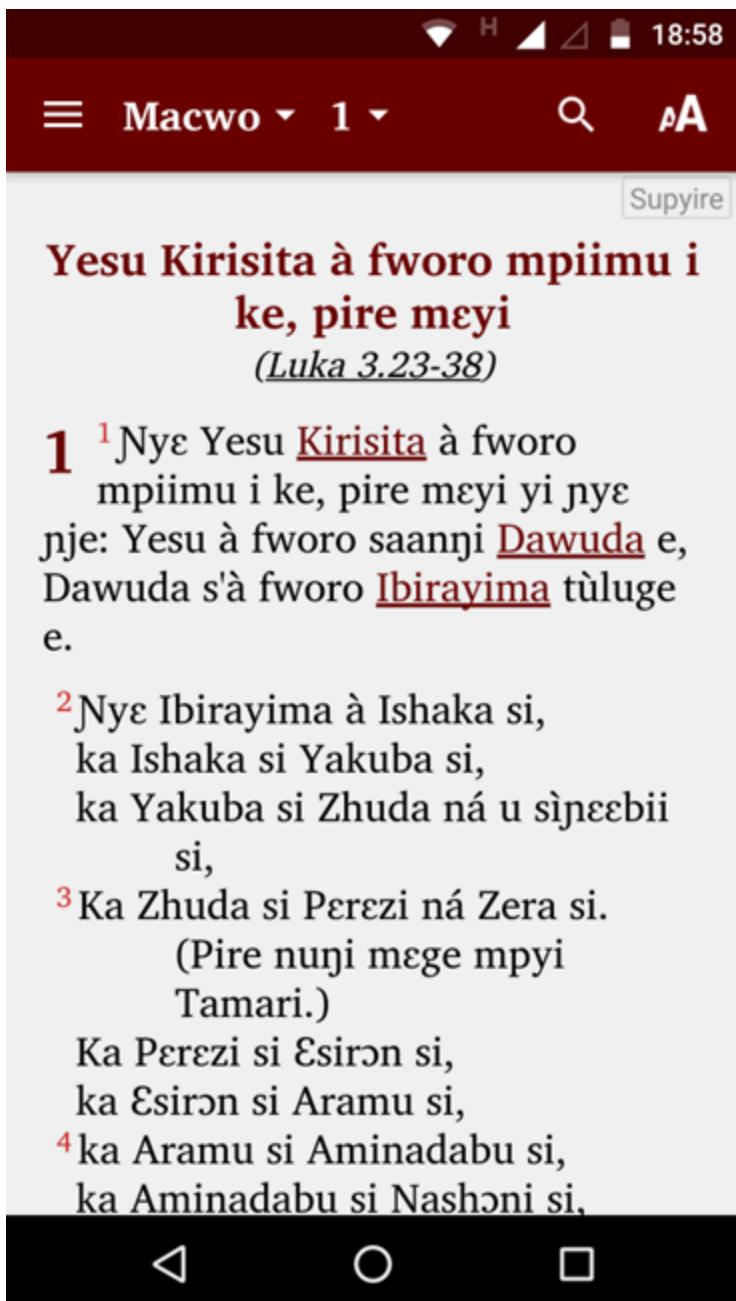
1. Go to **Tools → Settings → Build Settings → Gradle Cache Folder**.
2. Enter "C:\gradle" and OK.
3. Build an app. The Gradle cache files will be downloaded to C:\gradle.

Then, on computer B:

1. Copy the C:\gradle folder from computer A to C:\gradle.
2. Go to **Tools → Settings → Build Settings → Gradle Cache Folder**.
3. Enter "C:\gradle" and OK.

# 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*

## Yesu Kirisita à fworo mpiimu i ke, pire meyi

(*Luka 3.23-38*)

**1** <sup>1</sup>Nye Yesu Kirisita à fworo  
mpiimu i ke, pire meyi yi nye  
nje: Yesu à fworo saanji Dawuda e,  
Dawuda s'à fworo Ibirayima tùluge  
e.

<sup>2</sup>Nye Ibirayima à Ishaka si,  
ka Ishaka si Yakuiba si

**Ibirayima:** Izirayeli shiinbii  
ná dánafeebii tulyege ki. Kile à  
u pyi u fworo u yabiliŋi kìni i  
u a sì kìni uru Kile sí n-cyêe u  
na ke. Kile mú à jwɔmeeṇi  
lwó u á na uru sí lire kini kan  
u ná u tùluge shiinbil'á pi tèen.

ka Aminadabu si Nashoni si,



*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 Abə\k\*: Yawutuu shi jomč pu wo kafila kaŋa ka ki ŋε ke, ki kóri wu ŋε: «To», kelee «baa», kelee «baba». Pεεŋε mege ki ŋε kii, taanra ŋε kemu ni ge, lee na Yesu ya Kile yiri na «Aba» (Galasi Sheen 4:6).

\p \k Abəli\k\*: Abəli bi bye Kayən wo cuun. Wee w'a pye Adama ni Awa yε wo ja shuun wo we. Kile ŋii fyaara ni wu bi naari (Macoo 23:35; Luka 11:51; Heburuu 11:4).

\p \k Ibirayima\k\*: Kilε ya shi wemu shɔɔnri yaha wuyε mu ge, wee w'a pye Yawutuu pu wo shi we. Ibirayima w'a pye pu sefεlεε pu bεεri wo to n'a daa wu ni.

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

\c 1

\s Yesu Kirisa nɔhɔ tayirige sεmε

\r (Luka 3:23-38)

\p

\v 1 Yesu Kirisa nɔhɔ tayirige sεmε 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.

# 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.

<b>Option</b>	<b>Description</b>
<b>-i [filename]</b>	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.
<b>-a [filename]</b>	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.
<b>-f [fontname]</b>	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.
<b>-ic [filename]</b>	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.
<b>-l [lang-code]</b>	Set language for menu items and settings, e.g. en, fr, es
<b>-ft [feature=value]</b>	Set a feature, e.g. book-select=grid
<b>-vc [integer]</b>	Set version code, e.g. 1, 2, 3, or +1 to increment the current version code by 1.
<b>-vn [string]</b>	Set version name, e.g. 1.0, 2.1.4, or use +1, +0.1, +0.0.1 to increment the current value.
<b>-ks [filename]</b>	Set keystore filename. Use the full path of the file and enclose it in "double quotes" if there is a space in the path.

<b>Option</b>	<b>Description</b>
-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
```

# 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.

# 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.

## NOTE

*The subsequent sections follow the project menu structure in Scripture App Builder. Each main project menu topic has sub-categories for each sub-menu and tabs.*

# SAB Menu Bar

The menu bar runs along the top of the Scripture App Builder program window (Linux and Windows) or at the top of the screen (Mac).



# New App

This launches the **New App** wizard. See ***How to build your first app*** for more details.

# Open App...

This opens the Select App Project dialog box. To open an app project, find and select its **.appDef** file. This will located in the project's folder under App Projects. See *How SAB stores your project data files*.

# Open Multiple Apps...

This will display a list of your most recently modified app projects. Check the selection box for each one you want to open and SAB will load them all into the project tree view.

# Open App From Scriptoria

If your project has been uploaded to Scriptoria, choosing this menu item will allow you to enter the Scriptoria project URL, download the project and load it into SAB.

# **Close**

This will close the currently selected project in the project tree view.

# Save

This will save the current project, and is the same as clicking the Save button in the tool bar.

# Save All

This will save all unsaved projects.

# Copy App

This will create a mirror image of the current project, by default adding the word “**(Copy)**” to the project name to distinguish it from the copied project. This may be useful in creating similar projects but for different languages, or creating a test project to experiment with.

 **NOTE**

*The project copy **will have the same package name as the original**, so be sure to change it to avoid confusion or overwriting the original project installation.*

# View Project Folder

This will open the current project's project folder. This folder holds the project's .appDef configuration file and the project's data subfolders. See *How SAB stores your project data files*.

 **TIP**

*If you want to share your app project with someone else (i.e. someone who can help you test and find problems with your app), you'll need to give them your project folder and all it contains. They can copy it into their own App Projects folder, open it in SAB and use their own, temporary keystore to build, run and test your project. If audio files are necessary for your app and they are not stored in your project's audio subfolder, you will need to copy those separately.*

# View APK Output Folder

This will open SAB's output folder where it places any compiled APK files, ready for copying to a phone to test or share with others. You can change the default output folder under **Settings → Default Folders** on the SAB menu bar.

# View IPA Output Folder

This will open SAB's output folder where it places any compiled IPA (iOS) files, ready for copying to a phone to test or share with others.

# **View PWA Output Folder**

This will open SAB's output folder where it places any compiled PWA files, ready for copying to website.

# **Exit**

Use this to close Scripture App Builder.

# Build Android App APK

This menu item will initiate the compilation of the currently selected project, duplicating the functionality of the **Build Android App APK** button on the SAB tool bar. Note that multiple, device-specific APKs will be built instead of one, comprehensive APK if the **Generate device-specific APKs...** option is selected under the **App → APK → Multiple APKs** tab.

# Build Android App Bundle

This menu item will initiate the compilation of the currently selected project, creating an Android App Bundle (AAB) rather than an APK. An AAB file is necessary for publishing to the Google Play Store.

# Build Progressive Web App (PWA)

This will initiate the process for building a PWA. For details setting up the necessary components, please see ***Progressive Web Apps***.

# Build iOS Asset Package

For details on building an iOS Asset Package, please see **Installing and Building Apps on a Mac** under the SAB **Help** menu.

# Build Multiple Apps

In some cases it may be expedient to set multiple app compiles going at one time. This may take a long time, however, depending on available memory. Selecting Build Multiple Apps displays a dialog listing the currently open projects in SAB. You can check or uncheck the box next to each app project to select whether it is built. Use the drop down list box to choose whether you want to build APKs or Android App Bundles.

# Install Latest APK on Device

This duplicates the functionality of the **Install APK...** button on the SAB tool bar. If a phone is connected and set up for USB debugging, SAB will attempt to copy and launch the currently selected project's last built APK. See ***Installing the app on your phone***. If no APK is found for the project, SAB will prompt you to compile one.

# Export Data Safety to CSV

Publishing an app to the Google Play Store requires you to specify a series of answers to data safety questions. Selecting **Export Data Safety to CSV** will export a *comma separated values* file into the APK output folder. This file can be imported as part of the process of publishing.

# Settings

The Settings dialog box allows you to modify the SAB build settings and folders.

# The Code Libraries tab

## JDK Location

The Java Development Kit is a required set of code libraries for building apps. The normal SAB installation process should prompt you to download and install the JDK, and the **JDK Location** field should show the location of the installed files. If for any reason this is blank, or you need to change the version of the JDK, you can either **Browse** to another version already on your computer, or use **Install JDK...** to download and install a new version.

## App Builder Location

This is the folder in which SAB was installed. If you need to install a previous version for test purposes, you can download it from the **Previous Versions** section at the bottom of the [SAB download page](#) and Browse to it here.

# The Android SDK tab

## Android SDK Location

The Android Software Development Kit is a required set of code libraries for building apps. The normal SAB installation process should prompt you to download and install the Android SDK, and the **Android SDK Location** field should show the location of the installed files. If for any reason this is blank, or you need to change the version of the SDK, you can either **Browse** to another version already on your computer, or use **Install Android SDK...** to download and install a new version.

## Android SDK Packages

If installed correctly the four SDK packages should have green version numbers displayed (see example below).

**Android SDK Packages**

You need four packages installed for the Android SDK. If any are listed as 'Not Found' below, click 'Install Android SDK...' above to install (or reinstall) the required Android SDK packages.

Tools	<code>cmdline-tools 12.0</code>	<a href="#">Check Installation</a>
Build Tools	34.0.0	
Platform Tools	35.0.1	
Platform API	Android 14 (API 34)	

Click on **Check Installation** to make sure the four SDK packages are installed. If any of the four fields are listed as 'Not Found,' click **Install Android SDK...** to install or reinstall the packages.

# The Defaults tab

Currently, the Defaults tab has one field...

## **Start of Package Name**

Use this field to enter the default beginning parts of a package name that you generally use. See ***How Should I choose the app package name?*** for more details.

# The Default Folders tab

## **Default App Projects Folder**

Use this field to change the default location where SAB looks for app projects.

## **App Output Folder for Android Apps**

Use this field to change the default location where SAB places your compiled APK files.

# The Export Folders tab

## **HTML Output Folder**

Use this field to change the default location where SAB places your exported HTML files.

## **EPUB Output Folder**

Use this field to change the default location where SAB places your exported EPUB files.

## **PWA Output Folder**

Use this field to change the default location where SAB places your exported PWA files.

# The Build Settings tab

## Offline Mode

The actual program that compiles your project to create an app installer file is named Gradle. The first time you compile your app, Gradle will need to download some additional files to run. From that point on, it will try to check from time to time if more updates are required. If you do not have internet available after the initial extra download, check the **Use compiler offline** box. You may, however, need to uncheck this when you have an internet connection to get any updates.

## Memory Use

Normally, you do not have to modify this value. The accompanying explanatory text explains when you may need to change it.

# The Build Folders tab

## **Gradle Cache Folder**

Normally, you should leave this field blank to use the default cache folder unless you have a specific reason to choose another location.

## **Build Folder**

Normally, you should leave this field at the default folder unless you have a specific reason to choose another location.

# The After Build tab

You can set SAB to automatically install and launch an app after it is compiled if your phone is attached via a USB data cable and set up for USB debugging. See ***Installing the app on your phone***. You may need a 3rd-party app to determine your phone's architecture.

# The Interface tab

## User Interface

Use the User Interface dropdown to select the default user interface language for your version of Scripture App Builder. You can also change this using the interface language selector at the far right of the tool bar.

## Theme

Select the theme for the SAB interface.

### Dark Theme

Select the dark theme for the SAB interface. This will be used if your computer has its dark theme mode enabled. If you want SAB to use the normal theme, even if the computer's dark theme mode is active, select **Always use the main theme**.

# The Files tab

## Audio Files

Audio files can comprise some of the largest project data files. Use the setting here to tell SAB whether to keep audio files in their source location or copy them to the project folder's *audio data subfolder*. This second option will make it easier to save or share the project, but will double the size the audio files will take up on your computer.

## Fine Tune Timings

As explained in the tab text, SAB will automatically look in the specified folder for saved timing files and copy them to the project. You can modify the location if necessary.

# The Slideshows tab

For help with this tab, see **Creating Slideshow Videos** under the SAB **Help** menu.

# Create New Keystore

A keystore is a digital certificate file that ensures the app is legitimate. An app must be signed in this way so that it can be installed on a phone. You may use an existing keystore or follow this wizard to create a new one. This process is also included as part of the “New App” wizard.

Enter a new filename for the keystore, such as “keystore1.” Specify a password. Click **Next** to continue.

Enter an alias name for a key to create within your new keystore, such as the default “key1”. Specify a password, which can be the same as the password you entered on the previous page.

 **NOTE**

*It is very important to securely store the password for your keystore. Once the app is published, it cannot be updated without the keystore password. Make sure someone else in your organization has access to the app keystore passwords.*

Click **Next** to continue.

On the **Certificate Issuer** page, provide details of your organisation in at least one of the fields. Click **Next** to continue.

A new keystore will be created for you. Click **Close**.

# Install JDK

This menu item duplicates the **Install JDK** feature under the Tools → Settings → Code Libraries tab.

# Install Android SDK

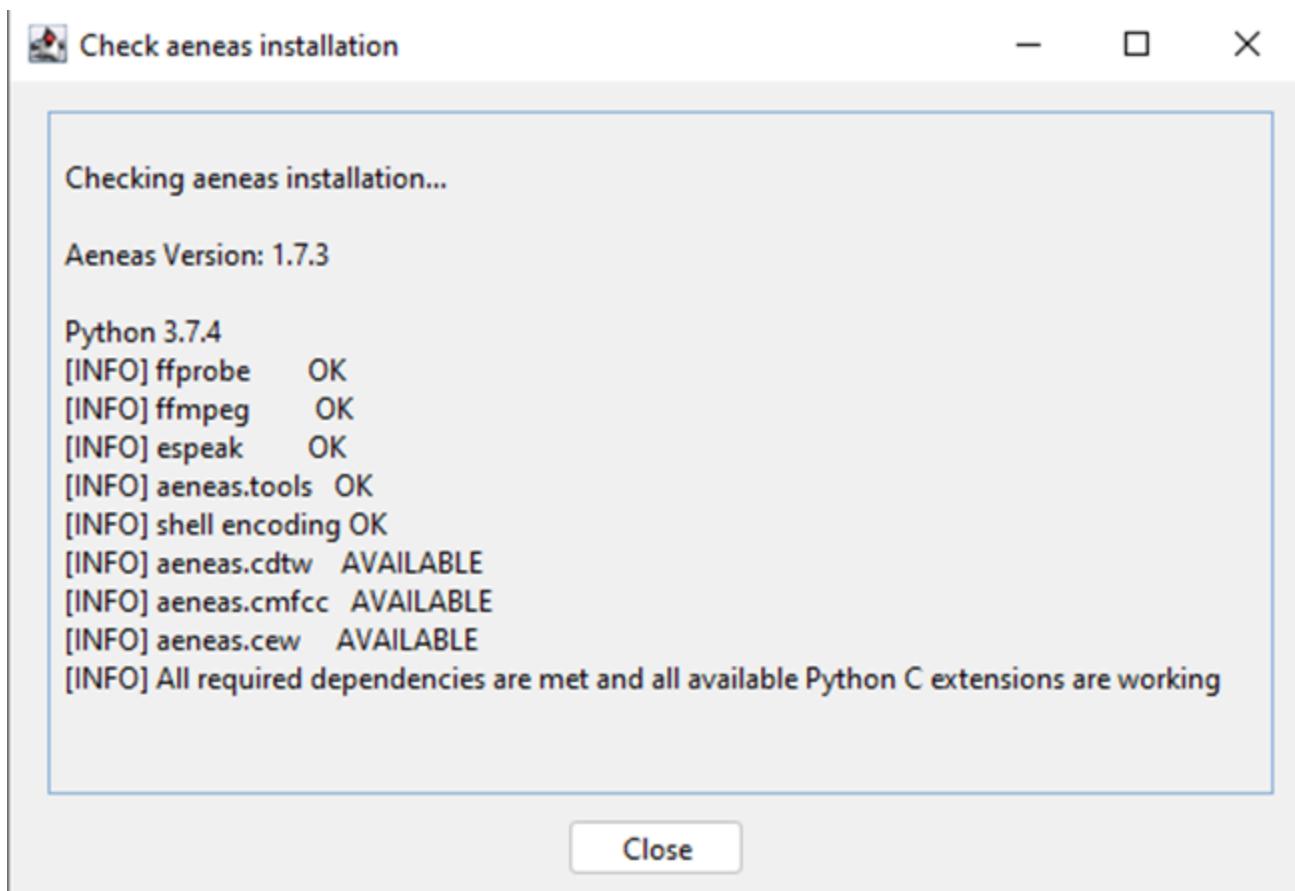
This menu item duplicates the **Install Android SDK** feature under the **Tools → Settings → Android SDK** tab.

# View Logs

As SAB builds an app APK file, it opens a window that displays the internal steps it is taking as it compiles the app. Once you close this window (when the app compilation has finished), you can still view the output or any error information in the View Logs window. There are two tabs—one for any error information and one for the compilation output. These are useful if you need to figure out why a build failed and/or can be sent to a technical specialist for help.

# Check aeneas installation

Selecting this menu item runs a process that checks that aeneas was installed correctly. This can help in troubleshooting problems and/or can be sent to a technical specialist for help. Below is an example of a successful installation check:



# The Help menu

The Help menu gives access to a number of helpful features, documents and links.

# Find Settings and Features

Use the **Find Settings and Features** dialog box to search for app settings you wish to set or modify. Start typing any keywords in the **Find** entry area and possible settings will display below. Double-click on a setting to open that location in SAB.

This feature can also be accessed using the **Find...** button on the right side of SAB's tool bar.

# Help Documents

Following Find Settings and Features are a list of helpful documents. Clicking on an item opens a PDF file that you can save locally if you wish.

# Community Website

SIL maintains a community website where users can post questions, comments or feature requests. The app builder community is quite active and members like helping each other with specific needs. If you have a question or are trying to do something in an app that is not clearly specified in the SAB help documents, searching for a comparable issue in the community website is a good first step.

# About App Builder

The SAB About box shows the current version and build number, copyright information and a link to the Scripture App Builder website.

# The SAB Tool Bar

The SAB tool bar is a series of buttons directly below the menu bar.



New App... | Save | Save All | Build Android App | Install APK... | Build PWA | Run PWA

# New App...

This launches the **New App** wizard. See ***How to build your first app*** for more details.

# Save

This saves the currently selected project.

# **Save All**

This saves all the unsaved open projects.

# Build Android App

This menu item will initiate the compilation of the currently selected project, duplicating the functionality of the **Build Android App APK** item on the **Build** menu. Note that multiple, device-specific APKs will be built instead of one, comprehensive APK if the **Generate device-specific APKs...** option is selected under the **App → APK → Multiple APKs** tab.

# Install APK

This duplicates the functionality of the **Install Latest APK on device** item under the **Build** menu. If a phone is connected and set up for USB debugging, SAB will attempt to copy and launch the currently selected project's last built APK. See *Installing the app on your phone*. If no APK is found for the project, SAB will prompt you to compile one.

# Build PWA

This will initiate the process for building a PWA. For details setting up the necessary components, please see ***Progressive Web Apps***.

# Run PWA

This will attempt to run the current project's compiled PWA. For details setting up the necessary components, please see ***Progressive Web Apps***.

# The App project menu

These are the details for all the tabs under the **App** menu in the project tree view on the left side of the SAB screen.

# App Name

The app name is what the user sees displayed on the phone under the app icon. Depending on the phone settings, the number of visible letters in the name may only be around 20 and may get truncated. Keep this in mind and keep the name short and descriptive.

You can use the **Translations...** button to set several app names, each based on the interface language of the phone (set under **Appearance → Interface → Languages**). Alternatively, you can just have one name in the actual target, vernacular language.

# Package

**The app package name needs to be unique and cannot change once the app is published.** 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.



## TIP

*You may need to plan ahead when choosing a package name. If you just have one Bible book to publish but are intending on adding more books to the app as they’re available, it’s better to add the word “Bible” than to put the name of the book. Similarly, if you might publish a text version of a single book AND a storybook, add a descriptive word to the package name e.g. org.wycliffe.senegal.wolof.john.storybook*

If you work for a Bible translation or publishing organization, 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.

# Project

The Project tab allows you to name or rename your project. The **Project Name** is not seen by the user, but it is important as SAB uses it to name the actual project folder on your computer (see *How SAB stores your project data files*). If you will be using a partner organization like Kalaam Media to publish your project to the Play Store, it is helpful for them if the project name starts with the language code and a hyphen e.g.

wol-Miracles of Jesus

You can assign a **Project Group** to nest similar projects together in SAB's project tree view. This might be useful if you are working on multiple languages. Creating a Project Group will also create a project group subfolder under your **App Projects** folder.



*If you modify a project name or group name later, you may have to close any affected projects in SAB, look in the App Projects folder and move projects back into their correct group.*

The **Project Description** is just for your own purposes and is not seen by the user.

# APK

APK stands for Android Package and is an app installation file. An APK file is the result of compiling an app in SAB and is the file that needs to be copied to a target phone to install and run the app. As such, the name of the APK is primarily only important to distinguish it from other files. If the user will receive the APK directly (via file transfer or WhatsApp message or something similar), and therefore will be able to actually see filename, the name of the APK should clearly reference the project or app name. Selecting the checkbox to **Append version name** (number) to the filename can also help with distinguishing between versions. If the app is installed from the Play Store, the user will not see the APK file.

The **Version code** is just an internal number that you can use within your project as you do internal updates.

# The Android Versions tab

The **Target Android Platform** refers to the current android platform that Scripture App Builder supports. This is the required platform supplied by Google who update this requirement about once a year. All new apps being published to the Play Store have to support this version, and existing apps will have to be upgraded within a few years depending on Google's specifications. New versions of Scripture App Builder are designed to keep pace with these requirements and simply compiling and republishing the app with the latest version makes the app compliant.

In contrast, the **Minimum Android Platform** can be specifically set to target a generation of Android phones using a particular Android version or later. In general, the older the platform (meaning the smaller the version number) the older the phone that can run the app. However, if your app uses a feature that only works on newer phones, this minimum platform will have to be set accordingly.

# The Install Location tab

You can specify here if the app is installed to a phone's external storage (i.e. a micro SD card) or its internal storage.

# The Multiple APKs tab

There are 4 types of Android phone architectures, and an APK file is designed to work on any of them. However, certain features (e.g. allowing saving as a video or the inclusion of GeckoView) can make the APK much bigger. Since these features depend on the phone architecture, smaller, device-specific APKs (known as Android App Bundles or AABs) can be built to target each architecture. The downside is that this prevents the sharing of the app bundle to another phone with a different architecture.

## NOTE

*To Publish to the Play Store you will need to create Android App Bundles (AABs). When a user installs an app from the Play Store, they receive the AAB associated with their phone's specific architecture, not a (universal) APK. To explicitly create AABs, go to the **Compile** menu in the main SAB menu bar and select **Compile Android App Bundle**. If you are using Scriptoria, you can create the AABs there. If you want to distribute your app on SD cards or other direct means to ANY phone, you will have to build an APK.*

# iOS

This tab is for building iOS apps. For more details on this, please see the document ***Installing and Building Apps on a Mac*** under the **Help** menu.

# App Signing

Your app must be “signed” to have permission to be installed on an Android phone. Signing means providing a way to authorize its installation by confirming that it is indeed created by you, the developer, and is not malware or a virus (or just an illegal app) in disguise. This is done by creating an encrypted, password-protected “certificate” file called a keystore.

If you don’t have one already, create a keystore by clicking on the **Create Keystore...** button which launches the **Create New Keystore** wizard.

 **NOTE**

*If you want, you can use the same keystore for all your apps. However, for various reasons, you may want to have a single, shared keystore for just a selection of apps—perhaps all the apps for a particular language. If you ever need to give your projects to someone else, they will need the password to be able to change the app.*

- i. Enter a new filename for the keystore, such as “keystore1.” Specify a password. Click **Next** to continue.
- ii. Enter an alias name for a key to create within your new keystore, such as the default “key1”. Specify a password, which can be the same as the password you entered on the previous page.

 **NOTE**

*It is very important to securely store the password for your keystore. Once the app is published, it cannot be updated without the keystore password. Make sure someone else in your organization has access to the app keystore passwords.*

Click **Next** to continue.

- iii. On the **Certificate Issuer** page, provide details of your organisation in at least one of the fields. Click **Next** to continue.
- iv. A new keystore will be created for you. Click **Close**.

Select your desired keystore and enter the password.

# Expiry

Normally, if your app is published to the Play Store or App Store, you don't want it to expire. If you want to update it, adding books or functionality, you can just publish the update and users will be able to get the new version. However, if you are creating a test version to distribute informally, you will probably want to set an expiration date.

There are three options on the Expiry tab for setting the app duration:

## **App does not expire**

This is the default and appropriate for published apps.

## **App expires on a specific date**

You can specify a date when all copies of the app will expire on all phones. This would be appropriate for a test version that you want to be in circulation for a limited time before getting feedback.

## **App expires a certain number of days after it is installed**

You can specify a number of days (up to 1000) that the user can access the app before it expires. Again, this might be useful for a test period.

You can also display an optional message when the app expires. There are two options here as well:

### **Show message and exit app**

This would be appropriate for ending a trial period and telling people to watch for the release version of the app, or perhaps to contact a certain person in order to give feedback. After showing the message, the app exits and cannot be run again.

### **Show message but allow the user to continue using the app**

This would be appropriate, for example, if the test version is accurate (consultant checking has been finished on the books) and the app has been distributed from phone to phone, especially if it is difficult for the user to get an updated version over the internet.

You can type the desired message in the text entry area at the bottom of the screen.

# Security

## Compression and Encryption

By default, SAB encrypts and compresses the data in the app. Normally, you should leave this setting enabled.

You can also choose to prevent users from taking screenshots in the app. A possible reason to enable this is if you have copyrighted art in the app and the artist does not want people to be able to copy their pictures without an associated copyright notice.

## Restricted Users

You can choose to let anyone install the app or restrict the installation in the following various ways:

### **Allow only selected devices to use this app**

This is the most restrictive setting and may be applicable in certain high-security contexts where only a very specific set of people should be able to install the app, based on their exact phone ID or serial number. Click **Specify Devices...** and add an entry for each phone you want to permit.

### **Require a device-specific access code to use this app**

With this option you can create a series of access codes to send to users so they can access the app. This requires one or more administrator accounts to be set up to send codes to users. This method allows a broader reach than just having selected devices set up to install the app.

### **Require each user to register with their details when they first use the app**

This option is useful for setting up a campaign directed at registering users and keeping them informed on updates and advertising things like listening groups to enhance Scripture Engagement in a community.

### **Show a calculator on start-up. Entering a code will give access to the app content**

This option is useful for high-security situations where the app needs to be completely hidden on a phone. A specific, long, digital number can be specified (e.g. 84567.294) as a code to put in the calculator's number field. Then hit = to transform and open the app.

 **NOTE**

*It is advised to make the app name and icon correspond to those of a calculator app.*

# The About project menu

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.

# 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.
- Copyright information for the app itself.
- Link to your website
- Link to a privacy policy
- Contact information, such as an email address or phone number.
- A “brand” image for your local organization and/or the icon image for your app.

# Which formatting codes can I use in the About page?

You can use the following formatting codes:

## CAUTION

Docu-notion does not allow for html tags to appear as text. Replace all { } with angle brackets

<b>Bold text</b>	Surround the text you want in bold with {b} and {/b} markers. Example: {b}This is in bold{/b}
<b>Italic text</b>	Surround the text you want in italics with {i} and {/i} markers. Example: {i}This is in italics{/i}
<b>Underlined text</b>	Surround the text you want underlined with {u} and {/u} markers. Example: {u}This is underlined{/u}
<b>Website links</b>	Use the format <a href="#">website</a>
<b>Email links</b>	Use the format [text] (mailto:address), where 'text' is the text to display and 'mailto:address' contains the email address. Example: Contact us by <a href="mailto:email">email</a>
<b>Phone numbers</b>	Use the format [text] (tel:number), where 'text' is the text to display and 'tel:number' contains the number to call. Example: Our number is [012-345-678] (tel:012345678)
<b>Image</b>	To add an image, first add the image file to the app illustrations ( <b>Images → Illustrations</b> ) and use the following HTML code:

	<pre>{img src="image1.jpg"}/}</pre> <p>You can specify width information, as a percentage (e.g. "80%") or a fixed number of pixels (e.g. "80px"), or in the <b>img.about-image style</b>:</p> <pre>{img width="80%" src="image1.jpg"}/}</pre>
<b>Left, right or centered alignment</b>	<p>To align text or images, surround them with the following {div} 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"}{img width="75%" src="img2.jpg"}/{div}</pre>
<b>Styles</b>	<p>To apply styles (defined in the <b>Styles</b> page):</p> <pre>{div class="q2"}This is styled as poetry{/div}</pre>
<b>Fonts</b>	<p>To apply specific fonts to text, first make sure that the fonts are specified on the <b>Fonts</b> page. Then use the following syntax:</p> <pre>{span style="font-family:font1;"}This is in font1{/span}</pre> <pre>{span style="font-family:font2;"}This is in font2{/span}</pre>
<b>Links within the About box</b>	<p>Set a marker to jump to when a text phrase is clicked. For example, set a language marker (Eng) to allow the user to jump to About box text in English—if you want to have multiple languages in the text.</p> <pre>{h3 id="Eng"}English{/h3}</pre> (sets the id <b>Eng</b> associated with this section) <p><pre>{a href="#Eng"}(Click here for the English version){/a}</pre> (marks the section to jump to)</p>

### TIP

*To center all the text in the About box, add “text-align: center;” to the CSS in the **body.about** UI style*

<b>To modify this...</b>	<b>Adjust this text style...</b>
About page formatting	body.about
Background block for an image	div.about-image-block
Image size / width	Img.about-image

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

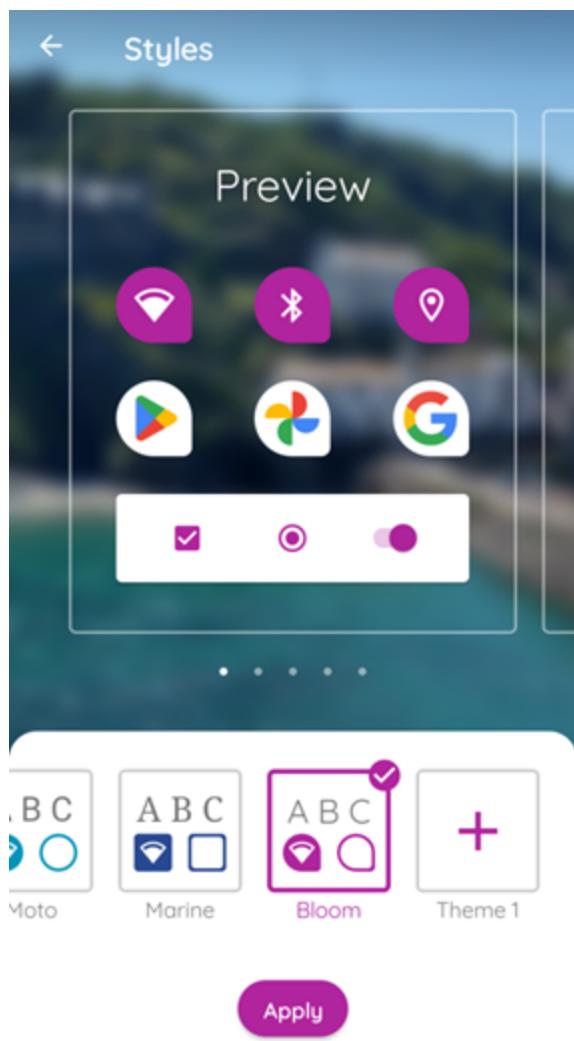
# Icons

Icons are important graphical elements of your app. In some sense, they represent your app to the user. Though SAB provides a gallery of icon images, it is recommended to put some prayerful, creative thought into creating a unique app icon, perhaps following a wider brand design for all your apps.

The **Icons** menu item covers both the icon the user sees on their phone screen which launches the app, and the notification icon associated with your app that appears next to any notification text messages that pop up. The **Icons** menu has three tabs:

# The Android Icon (Adaptive) tab

Devices from Android version 8.0 to Android 11 support Adaptive Icons. It is important to define an adaptive icon for your app, otherwise the icon will not be shown well on those phones. Foreground and background layers are specified separately. Users can decide which shape they want their icons, and the Android system will use the background layer to fill that shape, then place the foreground layer on top.



## TIP

*If your target phones include any Android versions earlier than 8.0, you will need to specify a Legacy Android Icon as well. It may make more sense to start with creating the legacy icon, add all the various sizes (which will also add a base adaptive icon) and then*

*finalize the adaptive icon. You can create all the sizes using a tool like **IconKitchen** (see Legacy icons below).*

You can adjust the foreground layer image using the percentage slider. Aim to have the image just a tiny bit larger than the circled area.

# The Android Icon (Legacy) tab

Legacy icons (those for versions of Android earlier than 8.0) need to be specified in a variety of sizes, and the phone's operating system will select which icon size that is best to display. Rather than creating all the sizes by hand, it is better to use a tool like [IconKitchen](#), which is linked to from this tab in SAB. IconKitchen will create a .zip file that you can download with all the various sizes. Use **Add Icon Images...** to select the downloaded zip and SAB will open the file and assign all the various sizes.

## TIP

*While IconKitchen has some basic tools to design an icon from scratch, you might prefer experimenting with using an AI picture generator like CoPilot or Playground AI to make a base icon image that you can then select into IconKitchen, adjust and download.*

SAB will create a series of folders under the project data folder's **Images** sub-folder, one for each icon size. (See *Project-specific data folders*). You can see these folder names under the **Filename** column of the Legacy icon display list. Each icon name is identical:

**ic\_launcher.png**

## NOTE

*If you want to publish your app on the Google Play store, you will need the 512 x 512 icon image stored in the **drawable-web** subfolder. You may want to use this image as part of the required Feature Graphic, too.*

# The Notification Icon tab

This tab is identical to the *Notification Icon* tab under **Engagement → Notifications**.

# Graphics

This menu allows you to set several optional graphical elements for your app. Click on a tab to modify that element.

# The Android Splash Screen tab

A splash screen is a title screen for your app that can automatically display for a specified number of milliseconds when your app launches. A splash screen can be an attractive “intro” to your app and should be set to display long enough for the user to read any text on it, but not so long that it becomes annoying.

There are three recommended sizes for Android splash screens. Presumably, the phone’s system will choose the one most appropriate for the particular phone’s resolution and aspect ratio. If one doesn’t fit exactly, we assume the OS will scale the nearest size to fit. In that case, for best results you should at least supply the largest size (720x1280).



**TIP**

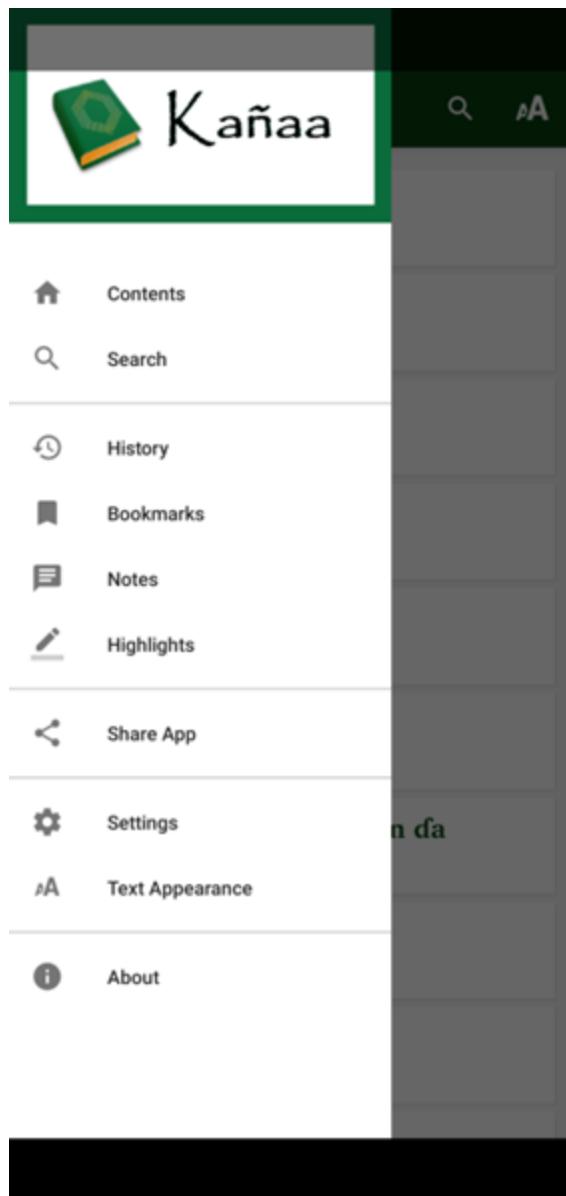
*If the Scripture text has already been published as a physical book, you might use the cover of that book as your splash screen to show continuity with the printed text. You could alternatively use a design incorporating your app icon with the app name. Try to harmonize the colors in the splash screen with the icon and interface colors.*

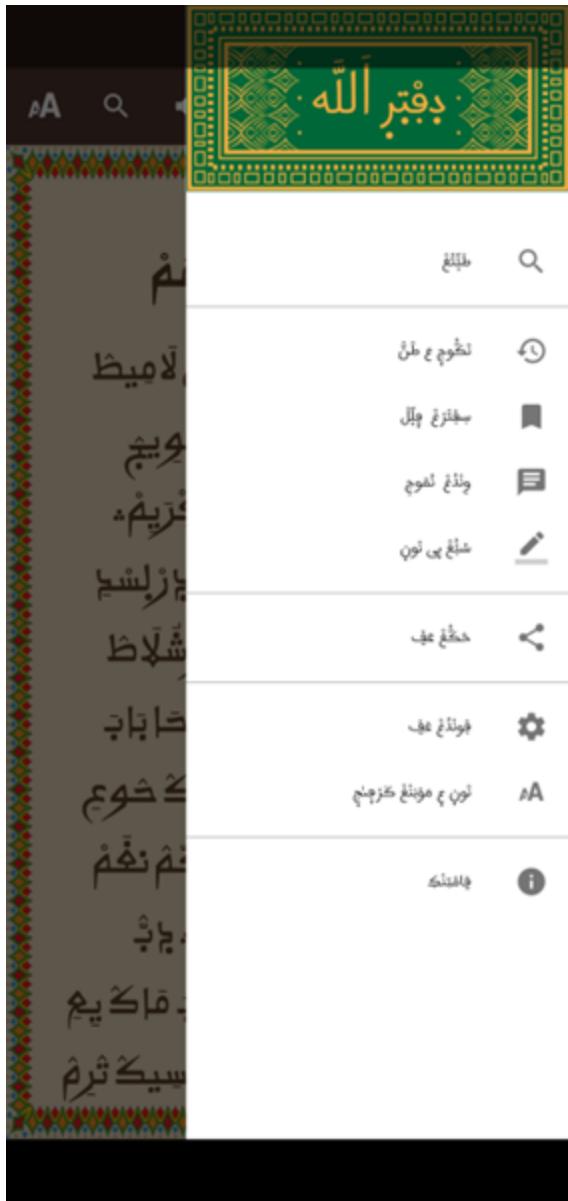
Check the box to **Show splash screen on start up**, and set the number of milliseconds. Around 800 – 1000 milliseconds is a reasonable, suggested interval.

# The Navigation Drawer tab

The optional navigation drawer image appears at the top of the app's slide-out navigation drawer. By default it is a simple, open book image. Adding your own image makes the app more aesthetically pleasing, especially if you use elements of the icon, interface colors and / or splash screen in the design. It can be a photo, your organisation's logo or any relevant graphic design.

As with the splash screen, SAB gives a number of suggested sizes, all using a 16:9 aspect ratio. If you want to include your own navigation drawer image, you should at least provide the largest size so it will scale well.





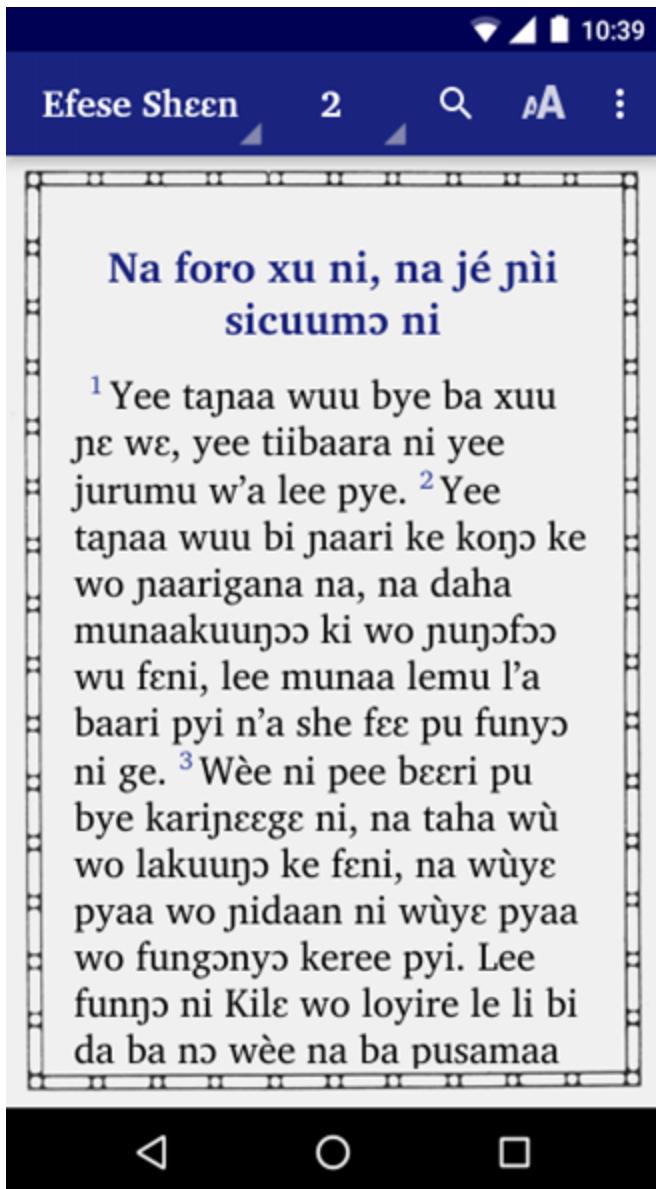
Example navigation drawer images

To modify this...	Adjust this interface style...
Navigation drawer background	ui.drawer
Navigation drawer item text	ui.drawer.item.text
Navigation drawer item icon	ui.drawer.item.icon

# The Borders tab

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

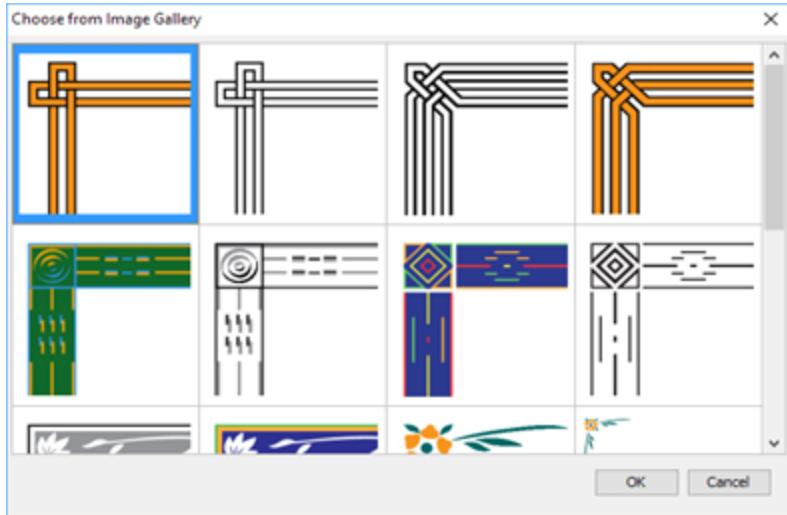




You can select a border from the gallery of border images supplied with Scripture App Builder, or you can create your own border.

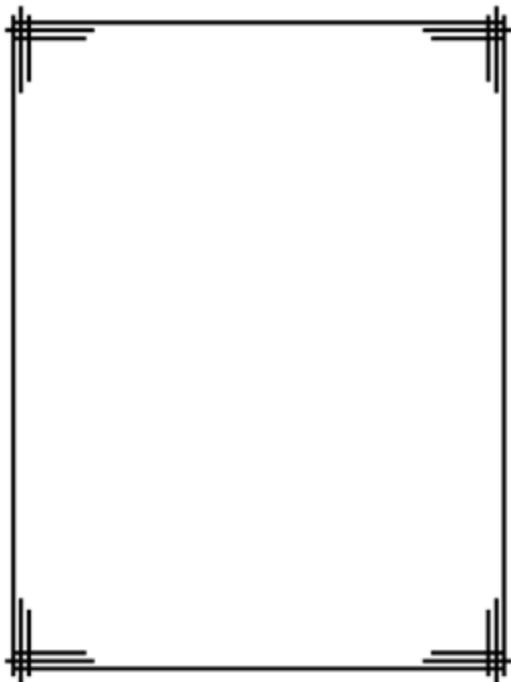
To select a border from the border gallery:

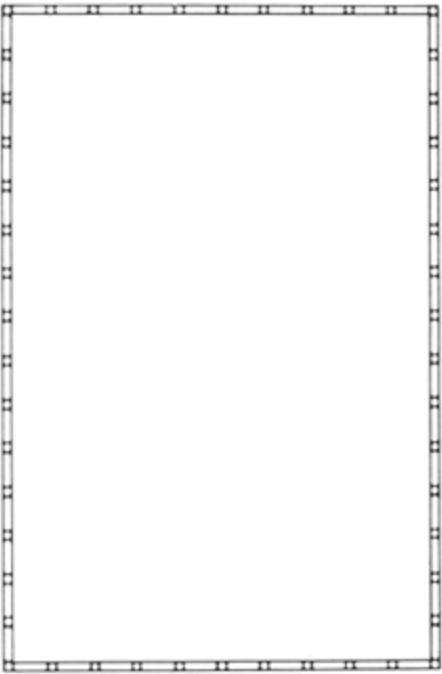
1. Click **Choose from Gallery...**
2. Select an image from the gallery and click OK.



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.

1. Go to the **Appearance** → **Graphics** → **Border**
2. Click **Add Image...** and select the border image.
3. Ensure that the checkbox **Include a border around the text in this app** is selected.
4. 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**

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 a toolbar displaying at the bottom.

# Interface

The Interface menu allows you to specify in which languages the user interface is available and how that translation is displayed. It has three tabs: Languages, Translations and Layout Direction.

# The Languages tab

The Languages tab allows you to specify which interface languages are available for the user to select from. This is best determined by knowing the app's intended audience and choosing languages that will meet their needs. NOTE: this does not refer to the actual language in which your Scripture text or other book resources are written.

SAB has a list of languages that already have interface translations, covering many of the world's national languages e.g French, English, Spanish and Chinese. You can choose which languages the user can select from when they access the app's settings. Use the **Add language...** button (see below) to add an interface language and provide the translation of all the terms.

SAB will detect what language the user's phone is set to. This is normally one of the national languages in SAB's list. At this point, you can specify what you want the app to do using one of two choices:

## **Use the current system language of the device. If no translations exists in this language, use...**

Select this option to detect the phone's current system language and try to match it with one of the languages you've provided. If the phone happens to have a language that is not in your list, you can specify which default language to choose.

### TIP

*Research where major communities in your target local language are living and what are the national languages in those areas. Provide those languages for them to choose from, and select one to be the default. For example, the Wolof people in francophone Senegal have diaspora communities in France and the USA. In this case, select French and English as available interface languages and perhaps select French as the default language if a user's phone isn't in English or French.*

## **Use this language:**

Use this option to force the app's interface to appear in a certain language, no matter what the phone may be set to. Users can still change the language to one of the ones you've

provided in the app settings.



**TIP**

*This option is useful if your app contains interface text or book resources (like Bible aids) in one particular national language. This would normally be if the specific audience are almost exclusively in just one country with one national language.*

## **Which interface languages do you want users to be able to choose from in the app Settings?**

This is the list of available interface languages. Choose all those that you think would help the app users, covering the countries where there are significant communities. Including a choice of languages does not significantly increase the size of the app.

Double-click a language name to specify the **Font**, **Size** and **Text Direction** (i.e. Left to Right) that the interface will display for that language.

### **Add Language...**

Use this button to add your own interface language. You specify the name, language code, default font and size, and the text-direction (for right-to-left languages). Once you select OK, the new language will appear already selected at the very bottom of the interface languages list. Click on the **Translations** tab to enter the translated interface element names for this language.

# The Translations tab

The **Translations** tab allows you to modify the names of interface items in any of the languages you selected to be available in the app. If you added your own language, it will appear in the right-most tab.

Use the **Export** button to export a text file with all the interface items to translate, in each of the available languages. This is useful if you want to give the list to someone else to translate. If you added your own language, you may have to enter its code and translation for each item. E.g. if I add the Wolof language and code “wol” (using the **Add Language** button on the **Languages** tab), then export the interface translations list using the **Export** button, I can add the Wolof translations. In the exported text file, each interface item has a heading (e.g. **\$ Menu\_Bible**) followed by a list of language codes and their translation for the **Menu\_Bible** interface element. At the end of the list for **Menu\_Bible** I can add a line **wol: Injil**. I can then do this for all the interface elements.

Once the list is finished, use the **Import** button to import the changes into SAB.

# The Layout Direction tab

The UI layout direction affects the position of the book name in the menu, and the side from which the navigation drawer slides out. There are four options:

- 1. Left to Right** – the default, with the book name displayed in the upper left and the navigation drawer sliding out from the left side.
- 2. Right to Left** – with the book name displayed in the upper right and the navigation drawer sliding out from the right side.
- 3. Use the direction from the current user interface language.** If the current language is normally written in a certain direction, the app will mirror that setting. This means that a user may change the app language in the settings and the app will respond with the correct UI direction.
- 4. Use the direction from the text being displayed.** This will depend on the book text being displayed in the screen area, or the top screen in two-pane view. See **Setting a specific book's text direction.**

# Colors

The Colors menu lets you customize almost any color in your app. The **Color Scheme** tab lets you set an overall look and you can further customize colors using the **Main Colors** and **Book Colors** tabs.

# The Color Scheme tab

SAB comes with around 28 pre-made color schemes. The color scheme affects the menu bar, bottom navigation bar icons, font sliders, book selection tabs and title text color. Click on the scheme you want and then choose which themes (**Normal, Sepia and Dark**) the user will have access to. These will appear under the text size (AA) icon on the menu bar.

# The Main Colors tab

The Main Colors tab allows you to customize all the named color settings for all three themes (Normal, Sepia and Dark). Double-click on a color entry to display the **Edit Color** selection dialog for all three theme. Use **Select Color...** to launch the color picker for each theme entry.

# The Book Colors tab

The Book Colors tab allows you to specify the tile background colors in the Grid View of the book selector (on the app's main menu). If your app does not have any Scripture books, this tab will be empty.

The Grid View groups Scripture books by category e.g. Pentateuch, Historical, Gospel, PaulineEpistle. You can specify the background color for each category and for all three themes. Double-click on a category to display the **Edit Color** selection dialog for all three theme. Use **Select Color...** to launch the color picker for each theme entry.

# Fonts

With the various tabs of the **Fonts** project menu, SAB allows you to specify one or more fonts to use in the app and to specify one or more keyboards and lexical models to aid users in typing in their language.

# The Main Font tab

SAB allows you to specify a main or “**common**” font for use in the app (by default). You can alternatively choose to specify a number of other fonts (using the **Font Files** tab) that will be embedded in the app for use with different books or book collections. If you provide several fonts, users will be able to choose between them in the Text Appearance popup and the Verse on Image editor.

# The Font Files tab

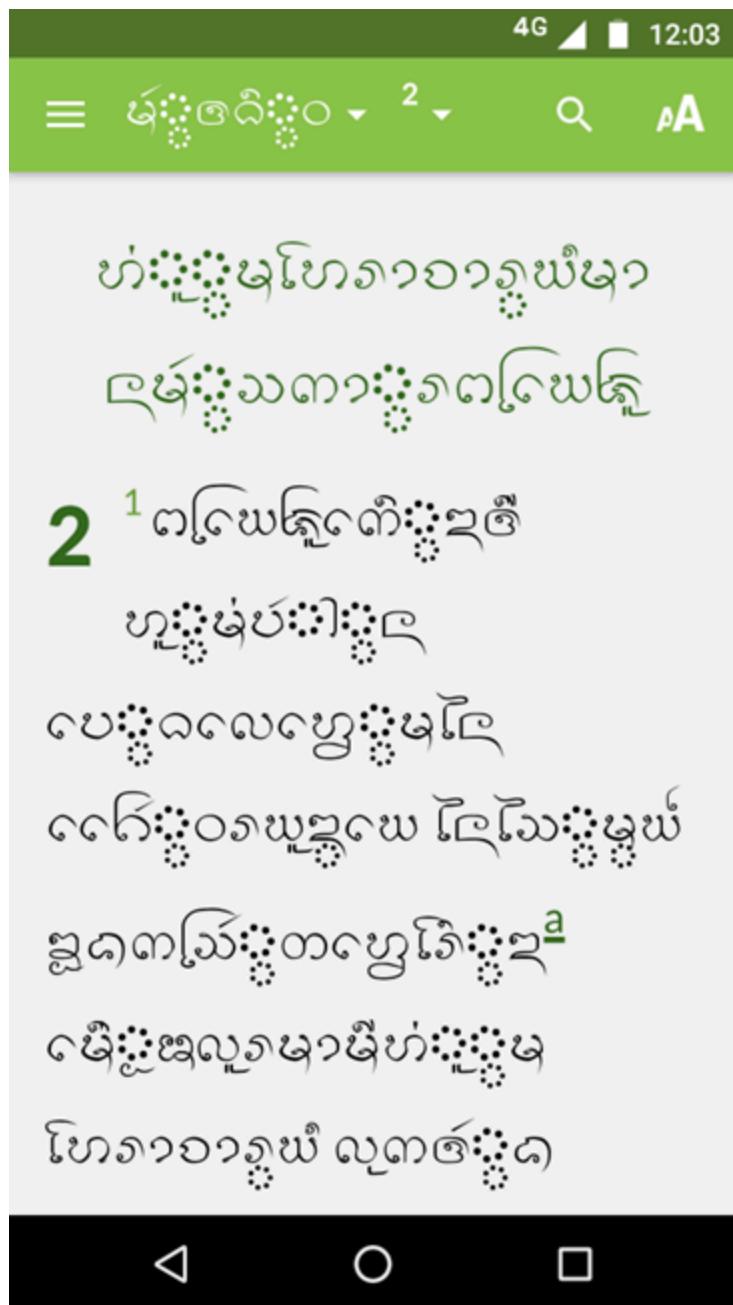
Use this tab to specify all the font files from which you can choose for the main font and fonts available under the **F**onts tab at the Book or Book Collection levels.

Click **Add Font...** and specify a reference “**Family name**” for the font. Then either select one of the suggested common SAB fonts or one of your own choosing. After clicking Finish, the font will be added to the list of available fonts from which to choose.

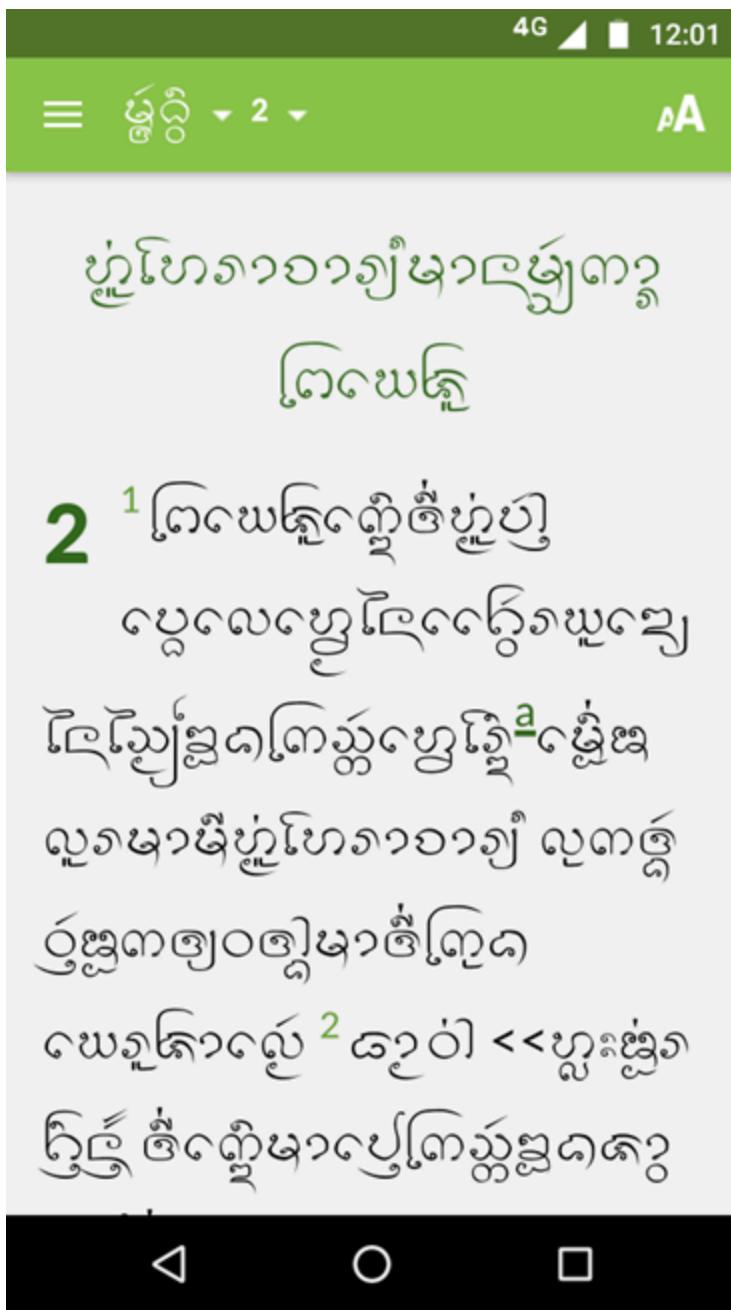
Right-click on a font file in the list to either delete it or edit its properties.

# The Font Handling tab

Use the Font Handling tab to activate the GeckoView font handling component. If you are using a non-Roman script or a Roman script with combining diacritics, it is possible that Android devices will not display your fonts correctly. To overcome these problems, try using the GeckoView library. GeckoView is a viewer component from Mozilla that replaces the standard Android viewer. 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** 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.

# The Keyboards tab

Use the **Keyboards** tab to specify one or more Keyman keyboards to include in your app. This allows users to use the keyboard for typing notes etc. in their language(s). The **Keyboards** tab includes links to [Keyman Developer](#) (for creating your own keyboards) and the [Keyman](#) website. It is worth checking the website to see if someone has already created the keyboard you desire.

**Important:** To allow the user to use a keyboard, make sure and select it at the Book Collection level on the **Styles** tab.

# The Lexical Models tab

Use the **Lexical Models** tab to specify one or more lexical models to provide predictive text for embedded Keyman keyboards. The **Lexical Models** tab includes a link to the [Keyman Developer](#) website for designing and compiling your own models.

# Styles

One of the powerful advantages of Scripture App Builder is the ability to make a fully-contextualized app for a specific target language and culture. To that end, SAB gives the app designer a lot of control over the text and interface styles. To see specific references and explanations to all the styles available in SAB, see **Appendix 3**.

Each style has multiple fields that can be customized, stored as underlying CSS fields. Styles are divided into two main categories: **Text Styles** and **User Interface Styles**. You can also create **Custom Styles**. Use the **Find** feature to search for a particular style to modify. If you know the USFM marker for the style, you can try using that for searching. For example, if you want to modify the style of a line of poetry, search for either “**poetry**” or “**q**” which is the USFM poetry marker.

Each style lists the **Name**, **Description** and **Designation**—the specific characteristics of the style. Double-click on a style to display the **Edit Style** dialog box. Use the **Style Properties** tab to modify the style and/or the **CSS** tab to add or modify specific style elements. Click **OK** when done. The modified style name will now be displayed in blue to show that it has been changed. To revert to the default values, right-click on the style and choose **Reset to Default**.

The **Appearance → Styles** menu is for setting “app level” styles. Styles can be set in 3 places: at the app level, the book collection level and the book level. By default, styles set at the app level apply to all collections and books. Some styles are only applicable to the app itself (e.g. menus, navigation drawer etc.). However, many styles, especially text styles, can be individually set for collections and/or specific books.

# The Text Styles tab

This is a broad category that contains many marker styles available in Paratext that are automatically transferred, and many other descriptive and functional elements in the app's content. These include different text categories like chapter and verse markers, poetry, tables, headers, footnotes, links, glossary words etc. Also included in the text styles are the various elements for quizzes (see the document ***Creating Applications with Quizzes***). Use the **Find** feature to search for a particular style. For more information, see **Appendix 3**.

# The User Interface Styles tab

Interface styles generally affect the user interface elements of your app. This includes the menus, icons, dialog boxes, tabs and navigation drawer. For more information, see [see Appendix 3](#).

# The Custom Styles tab

Use the **Add Style...** button to add a custom style.

# Setting styles at the book collection or book level

To change a particular book collection's styles, click on **Books** → **[Collection Name]** → **Styles**. The configuration area will display the text and interface style tabs with the elements that are applicable at the book collection level. You can also change a number of other attributes for the collection including the following:

- Text font
- Default text size
- Default line height
- Text direction (left-to-right / right-to-left)
- Language code
- Numeral system
- Verse numbers
- Keyboard

To change a particular book's styles, click on **Books** → **[Collection Name]** → **[Book name]** and then choose the **Styles** tab in the configuration area on the right.

# The Formatting tab

The **Formatting** tab gives specific control on whether chapter and verse numbers are displayed and how they are formatted. It also allows you to specify whether the words of Jesus can be displayed in red (if available in the current book) and whether they're in red by default. If turned on, this setting is available to be modified by the user in the app settings.

# The Text Size tab

The Text Size tab allows you to modify whether the user can adjust the text size in the app and to set the limits for changing the text size via “pinch-to-zoom.”

# Changes

You can define a set of regular expression “find and replace” changes to apply to the text in your source files. The changes will be applied during the app build process and the original source files will not be changed. For example, if you have added Paratext files from a Paratext folder that receives updates from your team and yet you need to modify certain text styles or markers to display correctly in the app, you should apply those modifications using the Changes feature. In this way, they will not be overridden by Paratext updates, or affect the Paratext project itself.

Changes applied at this level will affect the whole app. You can use the Changes tab *in the book collection* or book menus for changes at those levels.

Once Change items have been added, you can turn them off or on by clicking in the **Enable** checkbox. Right-click on a Change item to edit or delete it.

## Add Change from Gallery...

SAB comes with a limited set of Changes already defined. Click on the **Add Change from Gallery...** button, select any you wish to use and click OK.

## Add Change...

Click the **Add Change...** button to add a Change item. You can give it a descriptive name and specify the regular expressions to use for the Find and Replace fields.

# 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:

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

To set up analytics:

1. Select **Enable Analytics**.
2. Choose whether users can opt out of sharing user data
3. Click **Add Analytics Account...**
4. Choose an account type and enter your analytics account information.
  - For FCBH Analytics, you do not need any additional ID information.
  - For S3 Digest Analytics, you will need an S3 Bucket ID and an Identity Pool ID.
  - For Amplitude Analytics, you will need an API Key.

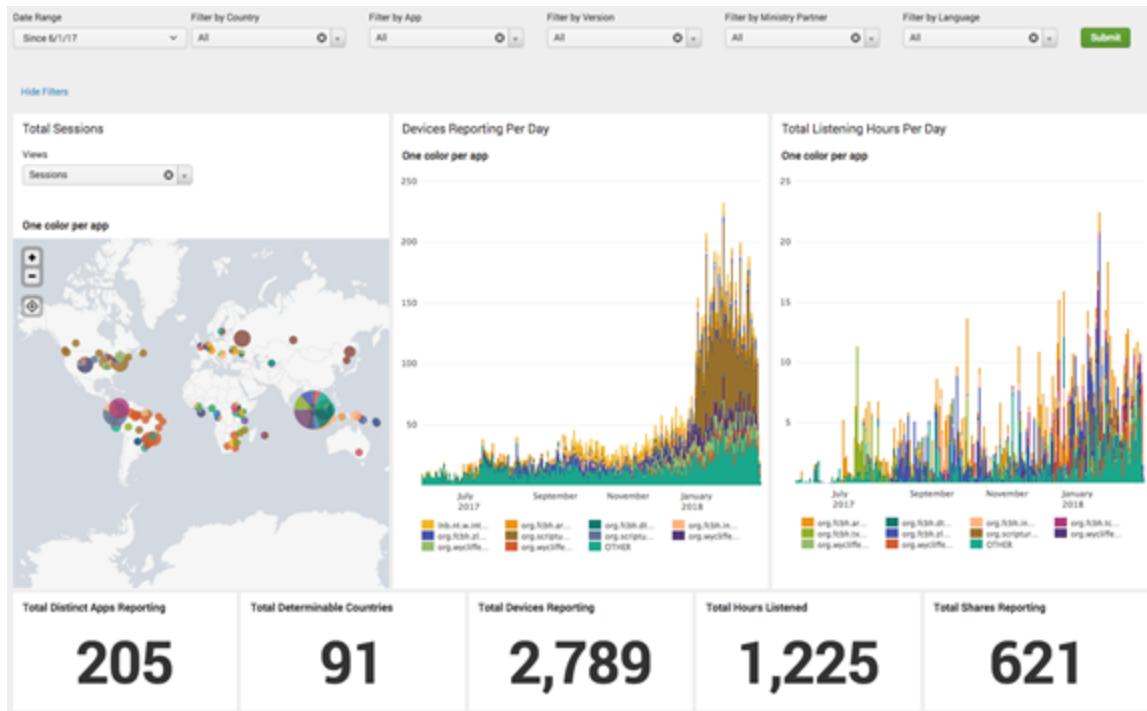
- For Firebase Analytics, you will need a google-services.json configuration file for your account.

# FCBH Digest Analytics

FCBH Digest Analytics is the most simple analytics service requiring almost no additional setup. 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](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](mailto: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).

# 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**

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**.

# 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.hndwtc.n2","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).

# 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.
4. 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**.
5. Highlight and copy this string into the API Key field in Scripture App Builder.

# Firebase Analytics

To configure Firebase Analytics, go to the **Data & Analytics** → **Firebase** menu (see below).

# Firebase

The Firebase menu allows you to configure multiple options available with a Firebase account. You can specify which options to enable using the Firebase Features tab, copy in your Firebase configuration file (.json) in the Firebase Configuration tab, and set up Push Notifications and In-App Messaging under their respective tabs, if so desired.

# The Firebase Features tab

## 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.

# The Firebase Configuration tab

Specify your Firebase Configuration file by clicking on **Browse...** and selecting your downloaded **google-services.json** file.

# The Push Notifications tab

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.

## 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 Features*.
2. In Scripture App Builder, go to the **Data & Analytics → Firebase** tab and enable **Firebase Messaging**.

## 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).

# The In-App Messaging tab

To set up the In-App Messaging feature:

1. Enable **Firebase In-App Messaging** under the *Firebase Features* tab
2. If you have not already done so, follow the instructions on setting up a **Firebase project**, described in the section above on *Firebase Features*.
3. Under the In-App Messaging tab, follow the instructions under **Configure In-App Messaging**.

# 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.

# Setting up User Accounts in Scripture App Builder

In Scripture App Builder:

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

# Setting up the database in 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 Features*.

## 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.

# The Data Deletion tab

Google's policies require apps with user accounts to provide a URL where users can request account deletion. See [Understanding Google Play's app account deletion requirements](#) and fill in the **Data Deletion URL**.

# The Publishing project menu

The Publishing menu helps you prepare your app for publishing to the Play Store, as a Progressive Web App or as an e-book (EPUB).

# App Store

Documentation coming soon. See the **App Publishing Checklist** and **Distributing Apps** PDFs under the SAB Help menu.

# **Web App (PWA)**

Documentation coming soon.

# E-books (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** 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.

"nɔɔ yilɔɔ Lagɔ 'na 'Denaade  
Galile -sеле- -ka me. <sup>15</sup> ɔ'ɔn 'ɔɔ:  
«Zle -ylɛ -ke lɔbhɔ "bhuc, "ne -  
Lagɔ 'na Kamale Zikpɛ "ne -ka  
la 'anye 'detlole-; 'zaa, 'a  
bhelia- 'anye 'kose -nyenye -wɛ  
'kɔ, -saa 'kele- Lagɔ 'na  
'Denaade 'dle "ku.»

## **Zeju 'la 'ɔnɛɛ gelenya 'na tetenoa**

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

<sup>16</sup> Zeju nɔŋo- Galile 'na -jinyɛ

"nɔɔ yilɔɔ Lagɔ 'na 'Denaade  
Galile -sеле- -ka me. <sup>15</sup>ɔ'ɔn 'ɔɔ:  
«Zle -yle -ke lɔbhɔ "bhuɔ, "ne -  
Lagɔ 'na Kamale Zikpε "ne -ka  
la 'anye 'detlole-; 'zaa, 'a  
bhelia- 'anye 'kose -nyenye -wε  
'kɔ, -saa 'kele- Lagɔ 'na  
'Denaade 'dle "ku.»

## **Zezu 'la 'ɔnɛɛ gelenya 'na tetenoa**

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

<sup>16</sup> Zezu nɔŋo- Galile 'na -jinyε

Readium

chrome-extension://fepbnnnkadjhjahcafoaglimekefifl/index.html?epub=filesystem%3Ac

Readium

Malke

Introduction  
Chapter 1

Zeu Klil 'na 'Denaade -ye  
Malke 'Wlewle 'kle- nele  
Zaa Batise 'na 'deseweli  
Zaa Batise 'nye Zezu  
batseme  
Sataa bea Zezu "me 'kose  
"ku  
Zeu 'cli- libli 3anç "ku  
Zeu 'la 'onç gelenya 'na  
tetenoa  
Zeu yamalea 'nyokpo -ye  
Zuzu -nyenye -ko keyi -we  
Zeu -yamalea Pieele 'na  
cucliccw  
Zeu -yamalea 'gwizinya  
'na 'kwleyia  
Zeu bli Galile 'na -gb  
"saa 'bhle 'Denaade  
Zeu yamalea  
cuclichqz'

Chapter 2

Zeu 'cu- -gbolabba  
Zeu 'la Levi  
Zeu bhule 'nue 'te well  
Zeu "ne -gbaje

Chapter 3

**'Wlewle 'kle- nele**

**Zaa Batise 'na 'deseweli**

*(Mt 3.1-12; Lk 3.1-18; Zaa 1.19-28)*

**<sup>1</sup> Dene Lagɔ 'Yowli Zezu Kli 'na**  
**'Denaade 'cile 'le "ku.** **<sup>2</sup> Weli -ye 'wa**  
**'kle- guckwɔmenyɔ<sup>a</sup>** Ezayi 'na "bɔgo  
"me ne le: «-Ame Lagɔ, 'e yi "neee  
lebhelebheyu -eyi 'yuku gelia-, -soc "ke -  
gwe 'na -yee "ke bhli -we -se.

**<sup>3</sup> 'Nyokpo 'bholo "kpale 'kose "ku "me, 'c**  
**'neee:**  
**«-A -se Anyinenyɔ 'na -gwe,**  
**-saa 'jee -ke "wli gbii.»**

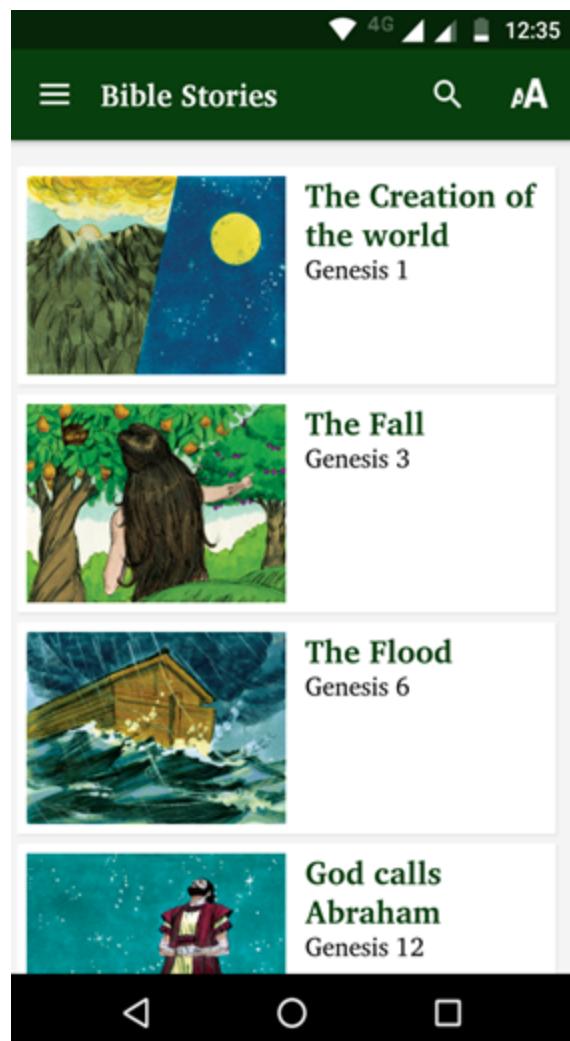
Viewing an EPUB Read Aloud document in the Readium Chrome extension

# 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. This can be an essential part of helping users engage with Scripture and locate other content in your app that otherwise might be hard to find e.g. front and back matter material, storybooks, a thematic index or glossary. For more information on setting the styles for the various contents menu elements, see

## **Appendix 3.**

The contents menu contains a list of images and titles and can have several levels as needed.



# The Contents Items and Screens tab

Contents menus are comprised of Items (clickable elements), headings (non-clickable items) and screens (groups of items).

## Contents Menu Items

1. 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 the book is not a Scripture book, you can still select which page to open. If you have multiple book collections, you can specify an associated layout.



*If the book does not have a linked book in another collection (i.e. the 2 pane view would not be able to show the same book in another collection), then choose single-pane view.*

- The **title** of the item, such as 'The Lost Sheep'
- An optional **subtitle**
- An **image** (optional) – to appear to the left of the title.



*Try and make your contents menu item images the same dimensions. Alternatively, you can set the **Image width** percentages to all be the same on the **Contents Settings** tab under **Contents Items Display**.*

2. You can reorder the items on the screen by selecting rows in the table and dragging up or down.
3. Go to the **Contents Settings** tab to specify display formatting and navigation options for the menu.



**TIP**  
By default the contents menu will show the item reference (e.g. chapter and verse). You can turn this off by unselecting **Show item references** in the **Contents Settings** tab.

4. Select the **Viewer** tab to see how the contents menu will appear in the app.

The screenshot shows the Scripture App Builder interface. On the left, a sidebar lists project files: 'Apps' (selected), 'cou-Bible en Wamey' (selected), 'App', 'About', 'Appearance', 'Data & Analytics', 'Publishing', 'Navigation' (selected), 'Contents Menu' (selected), 'Navigation Drawer', 'Bottom Navigation', 'Book Navigation', 'Deep Linking', 'Tab Types', 'Engagement', 'Media', 'Books' (selected), 'Wamey', and 'Français Louis Segond (Français)'. At the top, there are tabs for 'File', 'Build', 'Tools', 'Help', 'New App...', 'Save', 'Build Android App', 'Install APK...', 'Build PWA', 'Run PWA', and language selection ('EN'). The main area is titled 'cou-Bible en Wamey: Navigation > Contents Menu' and contains three tabs: 'Contents Items and Screens' (selected), 'Contents Settings', and 'Viewer'. Below the tabs, a note states: 'You can define a contents menu to be shown as the home page of the app. Users will be able to select books, stories and sections from a list of images and titles. You can place all of the menu items in the Home Screen or create additional screens as sub-menus.' The 'Viewer' tab shows a 'Home' screen with five items: 'Introduction' (with an image of an open book), 'La Genèse' (with an image of Earth), 'Nouveau Testament' (with an image of Jesus), 'Glossaire' (with an image of an open book), and 'Index thématique' (with an image of three people). On the right side of the viewer, there are buttons for 'Add Item...', 'Add Heading...', 'Add Screen...', 'Export...', and 'Import...'.

## Contents Menu Headings

You can add non-selectable headings or dividers in your lists of items in the menu. You may want to do this to group certain items under a header, or just to provide some decoration. To add a Heading, click **Add Heading...**



*To make a decorative heading, create a landscape image and set its width percentage to be 100%. Do not add a title or subtitle.*

## Contents Menu Screens

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

### TIP

*Grouping your items under screens can really help users find and access the content in your app, especially if you have different types of books and/or extra-Biblical content like front and back matter, glossaries, Bible studies etc. It is recommended to sketch out a proposed layout for your menus and then create the screens and items necessary.*

A screen is a group of items. Items on a screen can lead directly to books, or to other screens. By default a contents menu has one implicit screen called “**HOME**.”

To add a contents screen:

1. Click the **Add Screen...** button and give the screen a name.
2. You will need to link to this new screen from the screen a level above (its parent screen). Select the parent screen by clicking on it so it highlights. Click the **Add Item...** button for that parent screen, 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.

### TIP

*Make sure that the screen to which you want to add an item is selected by clicking to highlight it on the **Contents Items and Screens** page. Then click **Add Item...***

4. You can move items between screens by dragging item rows up or down.

# The Contents Settings tab

Use the Contents Settings tab to specify exactly how the contents menu works, including the title displayed (if any), what menu items are displayed and how users navigate between books and the contents menu.

## **Contents Menu Title**

Use this setting to choose what is displayed in the app toolbar while the contents menu is showing.

## **Contents Items Display**

Each item has a title, subtitle and a reference that shows which book chapter / page and verse the item is pointing to. You can choose which of these elements to display.

You can also adjust the proportion of the menu item image and text.

## **Navigation**

Use this setting to specify what type of navigation button is shown when a user is viewing a book.

## **Launch Action**

Use this setting to specify whether the app always launches to the contents menu, or to the last book viewed.

# The Viewer tab

Use the viewer tab to get an indication of how each screen in the menu will appear.

# Navigation Drawer

The navigation drawer is the menu that slides out from the left or right side of the screen, depending on the current *text direction* setting. This menu can also be accessed via the “hamburger” menu (three horizontal lines) in the menu bar.

# The Menu Items tab

As an aid to the user, you can add items to the navigation drawer that point to various, useful elements that can aid in helping the user go further in their studies or get answers to their questions.



*As part of a broader, Scripture Engagement strategy, you can add links to language websites, Facebook pages and WhatsApp groups.*

To add an item, click **Add Menu Item...** and choose from the list of possibilities. Be sure to add multiple translations of the menu item if you have multiple interface languages.

# The Settings tab

The **Settings** tab allows you to specify which app settings the user can modify. The list of settings is determined by which options you have enabled in your app. For example, the **Verse of the Day** notification setting will only appear in the list if you have enabled Verse of the Day under the **Engagement → Notifications** project menu.

Click Show next to each setting you want the user to be able to modify in the app. These will appear under the Settings menu item in the app's navigation drawer.

 **NOTE**

*To specify whether a setting option is turned on by default, see the respective setting's menu item in the SAB project menu. E.g. see **Engagement → Notifications → Verse of the Day** to turn it on by default.*

# The Sharing tab (sharing apps)

One advantage of a digital Bible over a printed one is that users can freely share the Scriptures with other people. SAB provides several, helpful ways to share an app with others. You can choose to allow the user to use any or all of them:

## **Share link to app on Google Play**

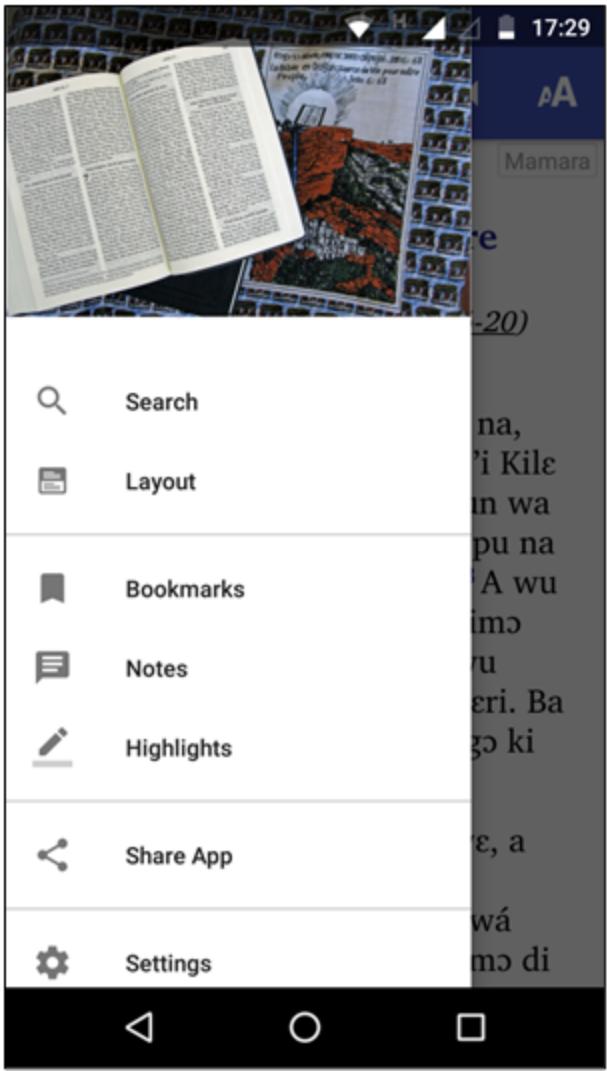
We strongly recommend publishing your apps to the Google Play store so that the diaspora can have access, even if your local context don't have internet access. If you have published it, then use this setting to provide an easy way to share the link via email or social media (e.g. WhatsApp).

## **Share link to app on website**

Enable this if you have your APK available to download from an external site (e.g. a language-specific website for a translation project).

## **Share app installer file**

Enable this to share your app's APK file directly 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.





SHARE LINK TO APP ON GOOGLE PLAY

SHARE APP INSTALLER FILE

CANCEL

5  
sip  
jor  
na  
for  
jé  
wo  
ko  
le  
ni, na sipyii pu kalaa.

<sup>4</sup>Ba Yesu ya xcjomc pu ni wε, a  
wu Simo pye: «Korogc ki fulo  
tacogonc ki ni, y'i yi jclcc pu wá  
lēhō ki ni fyaa pu feni!» <sup>5</sup>A Simo di



# The History tab

Enable the history screen so users can easily select previously viewed books and chapters. They can access the **History** list of saved locations under the app's navigation drawer menu. For information on setting the styles for various elements of the history feature, see [Appendix 3](#).

# The Search tab

The Search function allows users to find words across all book collections. There are several options you can set to specify how searches take place.

## Show Search icon and allow searching

If you enable this, the **Search** menu item will appear in the navigation drawer menu and the magnifying glass search icon will appear in the app menu, if there is room.

## 'Match whole words' is selected by default

You can choose whether or not users are restricted to searching for whole words.

## Show 'Match whole words' option on the Search page

This allows the user to turn off or on the 'match whole words' functionality.

## 'Match accents and tones' is selected by default

You can choose whether or not users are restricted to searching using accents and tones, if there are any in the language.

## Show 'Match accents and tones' option on the Search page

This allows the user to turn off or on the 'match accents and tones' functionality.

## Show special character input buttons on Search page

If you are allowing searching matching accents and tones, it is very helpful to include a list of any special characters (that would require an accent or tone marker) to be available for the user to select from.

## Specify the replacements and diacritics to remove...

If you are not requiring 'Match accents and tones,' it is important to specify which characters should have simple equivalents. You can either the syntax X > Y, with X being the accented character and Y being the non-accented equivalent (e.g. à > a) OR just specify the Unicode character number of the diacritic to remove.

# Bottom Navigation

The bottom navigation bar provides a quick way for users to access various elements of the app. It can be used in addition to or instead of a contents menu, but there is a limit of five items.

After selecting **Enable bottom navigation bar**, click **Add Button...** to add a button to the menu bar. For each button you can specify a name (with multiple translations if needed) and an icon. Once an element is added to the list in SAB, double-click on it to modify it.

← Matiye ▾ 1 ▾ 🔍 A 📖

## Itëfëta nte kerék Matiye panjare Yesu

### Vacérakélo Yesu

**1** Wëlin wamac vacérakélo Yesu  
Këris ale hnagiko hnënk ahnaŵ ale  
ŵaciko David, do David, umë dënk, hnë  
hnënk acérakélo fu Abëraham hna  
nagiko.

**2** Abëraham yeho rëm Isak ū. Isak yeho  
rëm Sakob ū. Sakob yeho rëm Yuda gë  
vëheryu<sup>a</sup>. **3** Yuda yeho rëm vëhni Peres  
gë Sera, hnëm hni ū Tamar yeho.<sup>b</sup> Peres  
yeho rëm Hesëron ū. Hesëron yeho rëm  
Ram ū. **4** Ram yeho rëm Aminadab ū.  
Aminadab yeho rëm Nason. Nason yeho  
rëm Salëma. **5** Salëma yeho rëm Bos,  
hnëmu ū Rahab yeho.<sup>c</sup> Bos yeho rëm  
Obed, hnëmu ū Rit yeho.<sup>d</sup> Obed yeho  
rëm Yese **6** ale yeho rëm ahnaŵ ale  
ŵaciko David ū.<sup>e</sup>

David yeho rëm Salomo ū, hnëmu ū  
Uri ryënkwëkawo ñëla. **7** Salomo yeho

There are six possible elements to link to:

## Contents

This displays the contents menu screen, if there is one.

## Bible

This will display the last, current book being viewed, or can display any book in any collection. When adding the button you can specify the target book to open or modify it later under the button's **Book** tab.

### TIP

*If you have any non-Bible books in your app, it is best to rename the **Bible** menu item to be **Read** or **Books** or something similar.*

## Plans

If reading plans are included in the app, this will display the reading plans selection screen.

## Search

If the Search function is enabled, this open the search screen.

## Settings

This will display the settings screen, allowing the user to modify the app settings.

## About

This will display the About box.

To modify this...	Adjust this style...
The bottom navigation bar background	ui.bottom-navigation.
The text under each icon	ui.bottom-navigation.item.text

<b>To modify this...</b>	<b>Adjust this style...</b>
The highlighted text of the current icon	ui.bottom-navigation.item.text.selected
The icon color	ui.bottom-navigation.item.icon
The highlighted color of the current icon	ui.bottom-navigation.item.icon.selected

# Book Navigation

The book navigation menu allows the customization of how a user selects and views books, and how Scripture references are handled.

# The Navigation tab

## Book Navigation

Books are most commonly selected by tapping on a book name displayed in the main app menu bar (the book selection popup). You can specify whether selecting a book automatically displays the chapter selector and then the verse selector.

If you turn off the book selection popup, you will need to provide another way to switch books e.g. a contents menu.

Books can be displayed using either a book list view or a grid view.





If the grid view is chosen, SAB uses the book's abbreviation as listed under the book's **Book** tab. The background colors can be set for the different groupings in the grid view (Gospels, Pauline Epistle etc.) using the **Appearance → Colors → Book Colors** tab.

Book selector in menu bar	<b>ui.selector.book</b>
Chapter selector in menu bar	<b>ui.selector.chapter</b>
Book, Chapter and Verse selector tabs	<b>ui.selector.tabs</b>
Background for drop-down book list view	<b>ui.button.book-list</b>
Background for drop-down book grid view	<b>ui.button.book-grid</b>
Titles of book groups	<b>ui.text.book-group-title</b>

# The First Book and Chapter tab

If there is no contents menu, you can specify which book and chapter is displayed when the app starts for the first time. You can select from any book in the primary book collection.



## TIP

*You may choose to open at a “Getting Started” or “Help” book to give the user help in learning how to navigate in your app. You can create this using the Word or SFM formats and add it to the primary book collection.*

Normally, you will want to open the app at the last book viewed, but you can specify here to always open at your specified First Book and Chapter.

# The References tab

These are set in Paratext or can be specified using USFM markers in any book file. Normally, selecting a reference causes a popup to show allowing the user to read a portion of the reference or jump to the source location. You can, alternatively, make selecting a reference open the source location full screen.

To modify this...	Adjust this style...
The cross reference popup body including background color	body.crossref
The reference popup title	span.ref
The reference background and text style	span.reflink

# 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.

# Setting up Deep Linking

To set up deep linking in your app:

1. Go to the **Navigation → 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/>

# 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>

# 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](https://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 Ø Deep Linking**
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\_abCdEF2GQ7ID9rLPlxk91khczmpGZl71

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>.

# Tab Types (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.



## Yesu xa moxo wanaani dañinin

(Matiye 15.32-39)

**8** Wo waxati kilinjo, jamaba kamarita tugun. Domolihen tun nte i bulu. Wo to, Yesu x'a la karandinjolu kili, a xo i yen xo : <sup>2</sup>« Nin moxo innu hinno be nte to. I tili sabo le mu bi ti, i be nte dahe jan, domoli hen si nte i bulu silan. <sup>3</sup>Nin n̄ x'i xonxoto bula i xa taxa i ya, i barako si dasa silo to, bawo dolu bodiya xa jan. » <sup>4</sup>A la karandinjolu x'a jaabi xo : « Alan si domolo bo min nin wulahin in xono, men si ninnu ha? » <sup>5</sup>Yesu x'i ñininxa xo : « Buurukun jalu be alu bulu? » I xo : « Worowula. »

<sup>6</sup> Wo kelinma, a x'a ñinin jama he i x'i sigi dugo ma. Wo xo. a





## Questions for discussion

1. Who would like to retell/summarise in their own words the Scripture passage/story we have just listened to?
  
2. What struck you particularly from what you heard?  
(something you liked, something you were surprised by, something that seemed hard to understand...)
  
3. What do we learn about God (or Jesus) in what we heard?
  
4. What do you think God is



*Text and Question tabs*

### To create a tab type:

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, see **Books** → [collection name] → [book name] → **Tabs**.

# The Engagement project menu

Engagement with the Scriptures is a critical element of Bible apps. It is important to prayerfully design your apps with the elements necessary to help users access and understand the Scripture resources it contains. SAB has a number of important engagement features to help in this regard.

# Annotations

Annotations provide ways for a user to deepen their personal interaction with and study of the Scriptures. If the user taps 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 they 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. For more information on settings styles for annotation features, see **Appendix 3**.

Users can synchronize their highlights, notes and bookmarks with another device. To enable this, please see the section on *User Accounts*.



## TIP

*If your book is not a Scripture text (i.e. does not have verses), by default you will not be able to tap on a sentence to highlight it. If you want users to be able to select the sentence, you will need to add \v verse markers to the book file but then hide the verse markers so they don't show. To do this, select the book in the collection and choose the **Features** tab. Use the **Show Verse Numbers** selector and set it to off.*

## Highlighting

Users cannot highlight titles or sub-headings, just verses of Scripture (or books with hidden verse markers). To highlight one or more verses, the user can do the following:

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

yàa na pa yirige lee yirigeg Mamara  
li na, <sup>15</sup> kɔnhɔ sipyaa sipyaa w'a  
dà wu na ge, wee di jìi  
sicuumɔ nixhɔbaama ta.»

<sup>16</sup> Bani koŋɔ ya taan Kile mu  
tehene baa, fo w'a ssɔ na wu Ja  
nigin pe wu kan na pye saraga,  
kɔnhɔ sipyaa sipyaa w'a dà wu  
na ge, wee ganha bu gyeegi we,  
ga wu jìi sicuumɔ nixhɔbaama  
ta. <sup>17</sup> Kile ya ta wu Ja wu tun  
na pa koŋɔ na, kɔnhɔ wu ba  
kiiri kɔn sipyii na we, ga w'a  
wu tun na pa, kɔnhɔ koŋɔ  
sipyii di shɔ wu baraga ni.

<sup>18</sup> Siŋva wemii hii dà wii na



yàa na pa yirige lee yirigeg Mamara  
li na, <sup>15</sup> kɔnhɔ sipyaa sipyaa w'a  
dà wu na ge, wee di jìi  
sicuumɔ nixhɔbaama ta.»

<sup>16</sup> Bani koŋɔ ya taan Kile mu  
tehene baa, fo w'a ssɔ na wu Ja  
nigin pe wu kan na pye saraga,  
kɔnhɔ sipyaa sipyaa w'a dà wu  
na ge, wee ganha bu gyeegi wε,  
ga wu jìi sicuumɔ nixhɔbaama  
ta. <sup>17</sup> Kile ya ta wu Ja wu tun  
na pa koŋɔ na, kɔnhɔ wu ba  
kiiri kɔn sipyii na wε, ga w'a  
wu tun na pa, kɔnhɔ koŋɔ  
sipyii di shɔ wu baraga ni.

<sup>18</sup> Sinvə wemū bù dà wuu na.



yàa na pa yirige lee yirigeg Mamara  
 li na, <sup>15</sup> kɔnhɔ sipyaa sipyaa w'a  
 dà wu na ge, wee di jìi  
 sicuumɔ nixhɔbaama ta.»

<sup>16</sup> Bani koŋɔ ya taan Kile mu  
 tehene baa, fo w'a sɔɔ na wu Ja  
 nigin pe wu kan na pye saraga,  
 kɔnhɔ sipyaa sipyaa w'a dà wu  
 na ge, wee ganha bu gyeegi we,  
 ga wu jìi sicuumɔ nixhɔbaama  
 ta. <sup>17</sup> Kile ya ta wu Ja wu tun  
 na pa koŋɔ na, kɔnhɔ wu ba  
 kiiri kɔn sipyii na we, ga w'a  
 wu tun na pa, kɔnhɔ koŋɔ  
 sipyii di shɔ wu baraga ni.

<sup>18</sup> Sipyaa wemu bu dà wu na,  
 Kile da ga ba kiiri kɔn weefɔɔ



To remove a highlight, a user can do the following:

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

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

### ⓘ NOTE

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.

To modify this...	Adjust this style...
Highlight background and text color for highlighter pen 1.	div.hlp1
Highlight background and text color for highlighter pen 2.	div.hlp2
Highlight background and text color for highlighter pen 3.	div.hlp3
Highlight background and text color for highlighter pen 4.	div.hlp4
Highlight background and text color for highlighter pen 5.	div.hlp5
The visual formatting of selected text	div.selected

## Notes

Using the **Notes** features, app users can add personal reflections, teaching notes, feedback on the translation, etc.

To add a note to a verse, the user can do the following:

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.

Mamara  
niyereye yaha, a namaa shuun  
fadeviire wuu pii di foro pu mu  
na pu fo. <sup>5</sup> A cèe p'i fya, na pu  
nuyc sogi sogi. A pee namaa  
shuun wu pu pye: «Sipyä we  
wu jne jnì na ge, na ha na yee  
d'a pa wee shaa xuu te ni we?  
<sup>6</sup> Wu wa na ha nige we. W'a jne

na foro xu ni. Ye wu bi yu yi  
mu na wu yaha Galile fiige ki  
ni ge, yi yi funyc kɔn yee na.  
<sup>7</sup> Wu bi yee pye na: «Li waha l'i  
waha, Sipyä Ja† na ba je  
kakuubyii keŋe ni, p'a wu kori  
tige na. Wu caxhugo caña  
taanri wogo w'a jne.» <sup>8</sup> Ba p'a

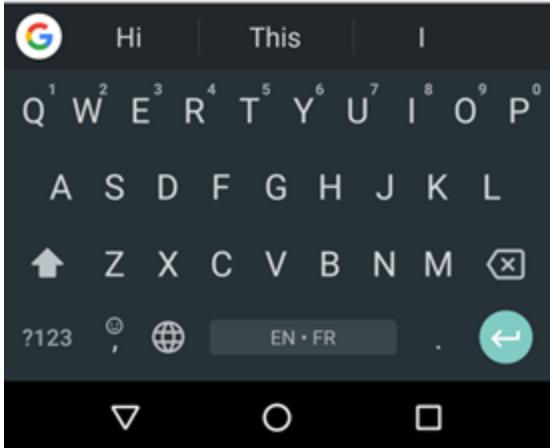


4G ▲ 10:13

← Add Note



ε œ œ œ



nixhugo ki ja we. <sup>4</sup> A pu hakilee di wuregi. Na pu niyereye yaha, a namaa shuun fadeviire wuu pii di foro pu mu na pu fo. <sup>5</sup> A cèe p'i fya, na pu juyc̄ sogi sogi. A pee namaa shuun wu pu pye: «Sipyá we wu jne jni na ge, jaha na yee d'a pa wee shaa xuu te ni we? <sup>6</sup> Wu wa na ha nige we. W'a jne na foro xu ni. Ye wu bi yu yi mu na wu yaha Galile fiige ki ni ge, yi yi funyc̄ kón yee na. <sup>7</sup> Wu bi yee pye na: «Li waha l'i waha, Sipyá Ja† na ba jé kakuubyii kej̄e ni, p'a wu kori tige na. Wu caxhugo caña taanri wogo w'a jne.» <sup>8</sup> Ba p'a

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

Users 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 under **Sharing Annotations**.

## Bookmarks

To add a bookmark to a verse, a user can do the following:

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.

daan Kile ni we.

Mamara

<sup>5</sup> «Ne wu ne erezen tige ke, yee di ne geye ye. Sipy a wemu w'a kori yaha ne wo karijneeg e ki ni, a ne be di gori yaha karijneeg e ni ni wufco ni ge, weefco wu da ja nagoo nijehemee pye. Bani ne bye ne ne ni yee ni we, yee da ja yafiin be pye Kile mu we. <sup>6</sup> Sipy a wemu bu bye wu ya sco na gori yaha ne wo karijneeg e ki ni we, wee na ba wa kpeen e na. Wu na ba bye ba erezen gen e ne we, p'a kemu kon na laha tige ki na ge, k'a waha. Yee gewaya



The screenshot shows a mobile application interface for the Mamara app. At the top, there is a blue header bar with the text "Yohana" and "15" followed by a dropdown arrow, a search icon, and a font size icon. Below the header, the main content area displays a verse in Mamara language. The verse begins with "daan Kile ni we." and continues with several lines of text. A red bookmark icon is visible at the end of the verse. The text is presented in a large, black, sans-serif font. At the bottom of the screen, there is a black toolbar with three white icons: a left arrow, a circle, and a square.

daan Kile ni we. Mamara

5 «Ne wu ne erezen tige ke, yee  
di ne geye ye. Sipy a wemu w'a  
kori yaha ne wo karijneeg e ki  
ni, a ne be di gori yaha  
karijneeg e ni ni wufco ni ge,  
weefco wu da ja nagoo  
nijehemee pye. Bani ne bye ne  
ne ni yee ni we, yee da ja yafiin  
be pye Kile mu we. 6 Sipy a  
wemu bu bye wu ya sco na gori  
yaha ne wo karijneeg e ki ni we,  
wee na ba wa kpeeng e na. Wu  
na ba bye ba erezen gen e ne  
we, p'a kemu kon na laha tige  
ki na ge, k'a waha. Yee gewaya  
yi na ba binne na yi le na ni,  
y'a sogi. 7 Yi bu gori yaha ne

To remove a bookmark from a verse, a user does the following:

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).

## Play Audio from Selected Text

If audio is present in the user's current chapter, enabling this allows a user to tap on a verse or verses and then tap on the Play button in the text selection toolbar. Audio playing will start at the first of the selected verses.

# Assistant (AI)

The AI Assistant provides Bible engagement help on a selected passage, such as a whole chapter or selected verses. You can set up the Assistant with a list of possible tasks, for example:

- “Give me some Bible study questions”
- “Share some devotional thoughts”
- “Suggest ways to apply this to my life”
- “Show how this passage fits into the message of the book”

These tasks are displayed in the Assistant tab. When the user selects one of them, the app connects to an AI provider, asks the question and displays the received response on the screen.



Mamara

## Marika 8

**How can I help?**

Give me some Bible study  
questions

Share some devotional  
thoughts on this passage

Suggest ways to apply this  
passage to my life

🏠  
Contents

📖  
Bible

📅  
Plans



*The list of tasks on the Assistant tab*



Mamara

## Marika 8

Share some devotional thoughts on this passage

### Reflecting on Mark 8

In **Mark 8**, we see Jesus feeding the four thousand, warning the disciples about the yeast of the Pharisees, healing a blind man, and Peter's confession that Jesus is the Christ. These events show us more about Jesus and our response to Him.

#### Jesus' Compassion and Provision

Jesus showed His compassion by feeding the four thousand, emphasizing His care for the physical needs of the people. This reminds us that Jesus not only cares about our spiritual needs but also our physical needs. He is our provider in all aspects of our lives.

#### Beware of the Yeast of the Pharisees

Just as Jesus warned the disciples about the yeast of the Pharisees, we are reminded



Contents



Bible



Plans



*The response generated by the AI provider*

To configure the AI Assistant:

1. On the **Assistant Tasks** tab, add some tasks, either by clicking **Add Task** or **Add Task from Gallery**. For each task you will need:

- **A task name:** This is the internal name for the task, not displayed to the user. Example: “*Bible Study Questions*”.
- **A task title:** This is the title of the task which is displayed to the user. Example: “*Give me some Bible study questions*”. Write the title in the languages of the app user interface. You can provide translations.
- **A task prompt:** This is the prompt to be sent to the AI provider when a user selects this task. It is not visible to the user. Example: “*Look at %reference% and give me six questions for a Bible study. These should be discovery Bible study style questions which encourage engagement with the text and discussion about the content, meaning and application.*”

When defining prompts, you can use the following variables:

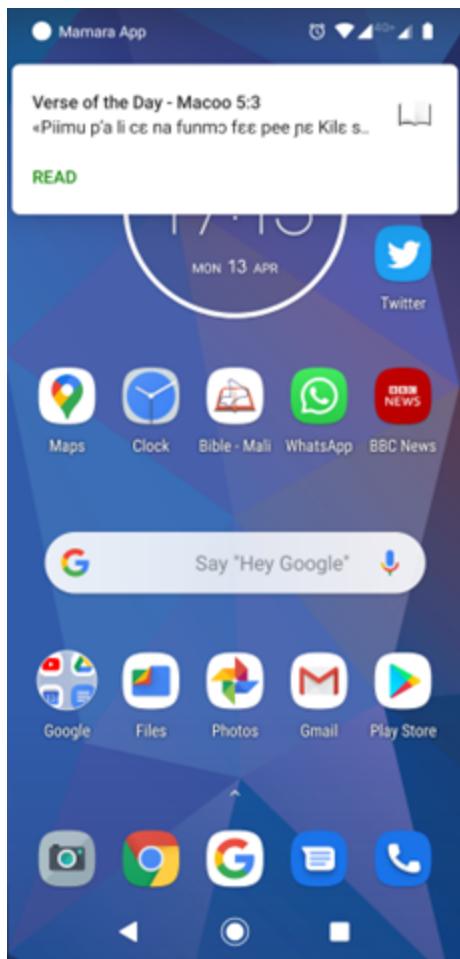
- **%reference%** - for the current Bible reference
  - **%book%** - for the current book name
  - **%chapter%** - for the current chapter number
  - **%biblical-language%** - for the current biblical language (Greek or Hebrew).
2. On the **AI Provider** tab, choose a provider from the possible options.
  3. On the corresponding **Settings** tab, e.g. “OpenAI Settings” or “Groq Settings”, enter the API key for the provider. You will need to have created an account with the provider for this, as well as specifying a billing method.
  4. Test the Assistant in the **Viewer**

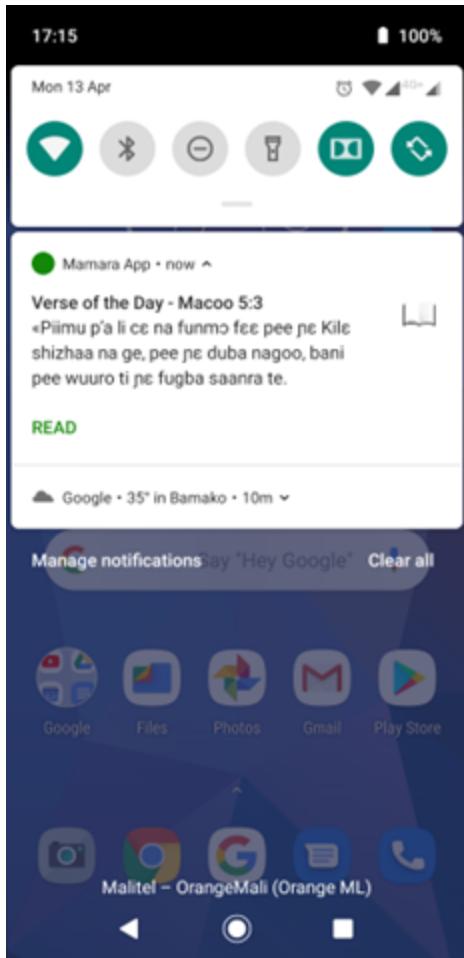
# Notifications

Notifications can be helpful for developing the habit of reading God's Word every day.

# The Verse of the Day tab

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.

To allow the user to have a verse of the day notification, do the following:

1. 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. You can allow the user to turn off/on verse of the day in the *app settings* under the navigation drawer.
2. 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.
3. Choose a **time of day** for the Verse of the Day to appear. Consider what default time of day in your cultural context might be the best for receiving the notification. The user will be able to change this in the *app Settings*.
4. Choose the appropriate **Christmas** and **Easter** date traditions.

5. 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.

# The Verses tab

Use the **Verses** tab to verify the list of verses from which the app will pull the verse notifications. SAB will create the list based on the actual Scripture books available in the app.

You may want to change a day's specific verse for a significant cultural event or holiday. To do this, double-click on the date and choose an alternate verse.

Use the **Special Dates** tab to modify which verses are displayed for the specified special dates.

SAB pulls verses from its **verses.txt** file, located under the Scripture App Builder installation folder (by default on Windows, the installation folder is **C:\Program Files(x86)\SIL\Scripture App Builder**) under the **info** subfolder.

Click **Rebuild Choice of Verses** to randomly reassigned verses for the year, pulling from the **verses.txt** file.

Click **Export...** to export the current list of verses in a simplified text file. You can edit this list and then use the **Import...** button to reimport it to SAB.

## NOTE

*If you modify **verses.txt**, it is liable to be overwritten next time you install an updated version of SAB. Similarly, if you modify a verse in the list on the **Verses** or **Special Dates** tabs or import your own verses, choosing **Rebuild Choice of Verses** will overwrite your changes.*

# The Daily Reminder tab

Your app can be configured to send a daily reminder to read the Bible. The reminder appears as a notification in the device's notification tray. When you tap on it, the app will open.

These notifications work offline. They are sent from the app rather than being sent over the internet. The user can turn on or off this feature and set their preferred time of day for the notification in the *app settings* screen.

By default, **Daily Reminder** notifications are not included in the app. To configure them in the app builder:

1. Select **Include Daily Reminder in this app**. If this option is not selected, a Daily Reminder will not be included in the app and the app user will not be able to turn it on.
2. If you want the notifications to be turned on by default, select **Turn on Daily Reminder 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.
3. Choose a default **Time of day** for the Daily Reminder to appear. The user will be able to change this in the *app Settings*.
4. You can customize the **Title** of the Daily Reminder and the **Message** that is displayed. You can add different text translations for any of the interface languages you have selected in **Appearance → Interface → Languages**.

# The Notification Icon tab

Click on the **Notification Icon** tab. You can create your own notification icon that will be used for both the Verse of the Day and Daily Reminder notifications.

Use the provided link to the **Notification Icon Generator** to easily create a notification icon. Download the zip package and use **Add Icon Images...** to select the zip file.

# Bible Reading Plans

Scripture apps can contain a selection of Bible plans, which include one or more daily readings. To add a plan, use the **Plans** tab. For more information on adjusting the styles of various elements of the plans, see **Appendix 3**.

# The Plans tab (for adding and modifying plans)

You can create your own reading plans, or add ones that someone else has created.

On the Plans tab, use the **Add Plan From Gallery...** button to add one of the plans that come with SAB.

Use the **Add Plan...** button to add a plan you or someone else has created.

Plans are defined in text files with the following 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:

\id	<b>Unique identifier for the plan (for internal use)</b>
\title	Title of the plan (seen by the user)
\descr	Description of the plan (seen by the user)
\img	Image
\day	Day in the plan (\day 1, \day 2, \day 3, etc.)
\ref	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.).
\heading	Heading to include before one or more references for a day.

If you need to translate a plan for different user interface languages, you can do this by adding the language code in square brackets after the standard format marker:

\id PLAN1

\title [en] Read through the Gospels

\title [fr] Lire les évangiles

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

\descr [fr] Lire les évangiles de Matthieu et de Marc, et apprendre à connaître Jésus, ce qu'il a fait, ce qu'il a dit et pourquoi il est venu.

\day 1

\ref MAT 1:1-17

Double-click the plan in the table to edit the name, description or image. Below are some screenshots of the Plans feature in action.

## Wərcəc cevəe ya pa na yìri Kile-nəhə na

**2** Yesu ya se Zhude fiige ki ni,  
Bətilehəmu ni, saannaa Hərədi†  
caŋa jii ni. Wu nizexhəgə ki na, a  
wərcəc cevəe pii di ba na yìri Kile-  
nəhə na, na nə Zheruzalemu ni. **3** A  
p'i yegeŋe pye na: «Yawutuu wo  
saan we w'a se nimə ge, mii wu  
wa wε? Wèe ya wu wəcə na, na  
wèe yaha Kile-nəhə ke na, na ba di  
ba wu pele.»

**3** Ba saannaa Hərədi ya yee logo  
wε, a l'i wu jaha wə fo xuuni, na  
fara Zheruzalemu sheen beeri na.

**4** A Hərədi di saraya jaha  
shəcnriyee juŋəfee†, ni saliya  
karaməgələcə† pu beeri pinne, na  
pu yege na: «Kirisa wemu wu jε  
Saan ni Izirayeli† wo Shəvcə wu  
ge, mii wu d'a yaa na se wε?» **5** A  
p'i wu cəl na: «Zhude fiige ki



Bible



Plans



← Plans

CHOOSE PLAN



**Mark in 16 days**  
16 days



**Proverbs in 31  
days**  
31 days



Bible



Plans

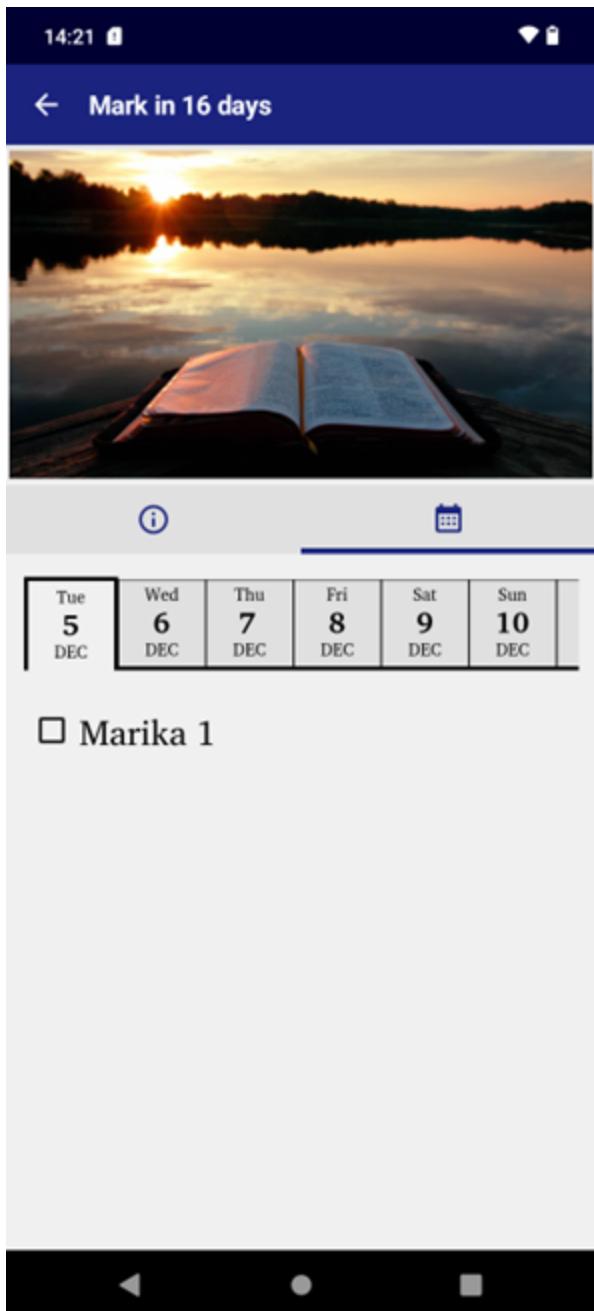


[← Mark in 16 days](#)**Mark in 16 days**

16 days

Read the Gospel of Mark, one chapter a day. This is an account of the life of Jesus, full of action. We learn about who Jesus is and why he came.

[START PLAN](#)



 **TIP**

*When naming the plan, don't put the number of days in the plan name as SAB will automatically put that in the subtitle. Don't forgot to use the Translations button to add titles and descriptions in all your interface languages if they do not already exist in the plan file.*

# The Plans Settings tab

Use the Plans → Settings tab to specify the default daily reminder times for plans.

# The Plans Viewer tab

Use the Plans → Viewer tab to get an idea of how the plan selection screen will look in the app.



## TIP

*You can easily generate reading plans using an AI tool like Microsoft CoPilot. Give it a plan as a model and then ask it to generate a new plan using that model. You should always check the output, however, to make sure the book IDs, chapters and verses are correct.*

# Verse on Image

Including the Verse on Image feature can be an important Scripture Engagement tool to encourage believers to share God's word with each other, or to evangelize non-believers. It can also be used to create Facebook, Instagram or other social media postings to increase interest in the language and/or in spiritual topics.

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:



*If your book is not a Scripture text (i.e. does not have verses), by default you will not be able to tap on a sentence to highlight it. If you want users to be able to select the sentence, you will need to add \v verse markers to the book file but then hide the verse markers so they don't show. To do this, select the book in the collection and choose the **Features** tab. Use the **Show Verse Numbers** selector and turn it to off.*

16:00

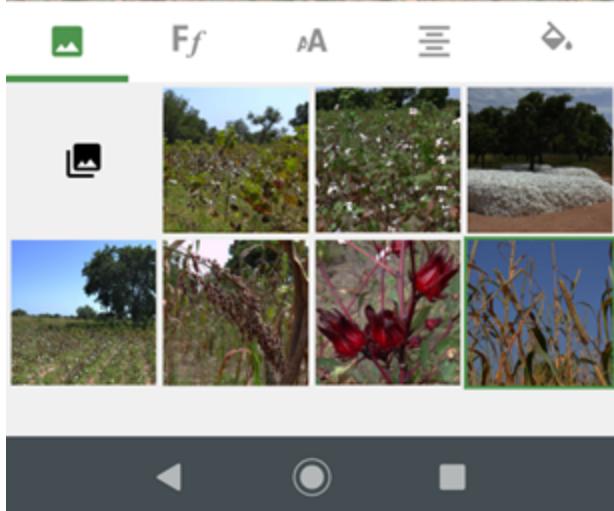
⌚ ▽ 4G+

← Image



Bani koŋɔ ya taan Kile  
mu tehene baa, fo w'a sɔɔ  
na wu Ja nigin pe wu  
kan na pye saraga, kɔnhɔ  
sipyaa sipyaa w'a dà wu  
na ge, wee ganha bu  
gyεεgi wε, ga wu jìi  
sicuumɔ nixhɔbaama ta.

Yohana 3:16





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)
- Text font
- Text size
- Bold or italics
- Letter spacing
- Line spacing
- Left, center or right alignment

- Text color
- Text opacity (transparency)
- 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.



# The Verse on Image tab

To activate the Verse on Image function in your app, check the box **Include the ‘Verse on Image’ editor in the app.**

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. To activate this option, go to the **Verse on Image** tab and check the box **Allow saving as a video, combining the text and audio.** This will require the image processing library, **FFmpeg**, to be included within the app.

 **NOTE**

*The FFmpeg library will add around 52MB to your APK by default. You can reduce the size of the APK by creating separate APK files for each phone architecture, including only the respective portions of the library in each one. See the Multiple APKs section under **App → APK**.*

# The Backgrounds tab

Users can either use one of the backgrounds included in the app or choose their own background from their phone image gallery.

As part of your Scripture Engagement strategy, try to include a set of backgrounds in your app that might appeal to the users and prompt them to share verses with others.

SAB provides several default images but you can add more in several ways:

Use the **Add Image...** button to add images that you provide. These can be photos that you, friends or colleagues have taken, or ones from a website. Just make sure you have the rights to use the images. The **Backgrounds** tab has links to several sites that provide free images, though you should still give credit in your About box.

You can also add images from the SAB gallery using **Choose from Gallery...**. Pay attention to the **File Size** of the images, as your app will increase in size with each one.



*Try to choose images that are relatively simple with an uncluttered center area so that the overlaid text can be easily read.*

We recommend using the **Crop each image to make it square** and **Resize the image files to reduce the app size** checkboxes. Pick a maximum width that reflects an average screen resolution of your target phones e.g. 720 - 1024 pixels.

You can also provide a choice of fonts for users to apply to the text in their images. To do this, go to the **Appearance → Fonts → Font Files** tab.

# The Watermarks tab

Use the Watermark tab to specify a small, PNG image that will be placed on all verse on image pictures. You can use this to “brand” your images so people associate them with your app, hopefully creating interest in installing the app itself and accessing God’s Word.