

**Accessing Scripture selections
on a smartphone or tablet
using icons without words
aka “Sliss”**

Geoffrey Hunt

Wycliffe Bible Translators
sliss.gh@gmail.com

Dr Maurice Manktelow

Member of MissionAssist
[New name for Wycliffe Associates (UK)]
support@sliss.info

October 2013

Contents

Introduction	1
User Concept.....	1
Technical Requirements.....	2
Setting up the user device	2
Introduction to technical Concepts	3
Visual Design concept	3
Tools required to create the system.....	5
Implementing the design	5
Listening to audio files and displaying text files	7
Important note on filenames.....	7
Case sensitivity.....	7
Self-explanatory names	7
Hiding folders and files	8
Distribution	8
Video formats	8
Icon size and formats.....	8
Frequently Asked Questions	9
Use on Android phones / tablets	9
Appendix 1: Setting up an Android Phone/Tablet for video files	10
Preparation	10
Setting up the device	10
Using the device.....	11

Introduction

Smartphones and tablets are increasing very quickly around the world. In the developing world huge numbers of people are using mobile phones, and these phones are becoming more capable with higher resolution screens and good audio. As a result, missions are adjusting their approaches to take advantage of this phenomenon.

Sliss is a method of making Scripture available on smartphones and tablets with access by icons that do not use any words or numbers.

For Deaf people, use Scripture videos. For illiterate people, use Scripture audio files. And for new readers, use either Scripture text files or Scripture audio files. [Note: Many first-generation literates find a New Testament or Bible is just too big; they need some way to access it in bite-sized chunks without requiring them to remember references.]

The purpose of this document, then, is to describe how to create and maintain such a system for smartphones or tablets.

Sliss is a ***work in progress***; it will continue to be developed.

The initial version has been implemented to display sign language videos on Android devices that have a slot for an SD card. Further development will focus on other popular devices, e.g. those from Apple.

The system has been devised with Biblical content in mind, but the same design concepts could be used for non-Biblical content.

Terminology:

- Because of the basic system being described here the word “passage” is used to refer to, but is not limited to, a Biblical passage.
- Similarly the word “video” could also refer to a sound file or a text file.
- Some devices are mouse sensitive – in which case the user will “click” an item. Other devices are touch sensitive in which case the user will “touch” or “tap” an icon. In order to avoid repetition, any one of these terms may also refer to the other two.

User Concept

The user will start the system by clicking on an icon¹ on their device to access the first level (level 0) of the SLISS icons. From this initial start page they will navigate to the passage they want to view via a series of icon links. It is envisaged that no more than 4/5 clicks will be necessary to get to the required passage.

For a short demonstration please see the SLISS website <http://slis.info> and follow the link on the demo page.

¹ The icon will need to be created by the user as either a clickable icon or a desktop shortcut icon – see later discussion with regard to specific devices.

The icons provided with the initial demo are not good examples of icons, but they serve the purpose of demonstrating the concept. As far as the authors are aware, there is no single set of icons available that can be used for this purpose². Rather, it is an opportunity for Christian artists to step up to the plate (to use an American idiom) and so provide sets of icons so that the Word of God can spread more easily throughout the world.

Developing icons for this purpose is a specialist endeavour which is discussed in a separate document titled “Icons for SLISS”.

Technical Requirements

The navigation system uses basic HTML and CSS files. Therefore, it will work with any web browser.

Sliss is **not an app**; rather it is a way of organizing data, so that it can be easily accessed. Sliss can be customized by a person with average IT skills; it just requires patience. Once customized, it can be easily replicated.

The system is designed to work on a microSD card inserted in a smartphone or tablet. It will also work on a standard PC. Technically, it could also function from an online website – but, if the user is trying to access videos, the large size of the files, the download speed and the cost could cause problems.

If a smartphone or tablet doesn't have an SD card slot, it may still be possible to make SLISS work, but the requisite research for this situation has yet to be done.

The video format is also a problem, because not all browsers support all video formats. The files for the demo were produced in the MP4 format using the H.264 codec, which is believed to provide the widest support.

If the system is to be accessed over the Internet, an additional consideration arises for video formats, namely how they are indexed. Normally the index is at the end of the file and therefore the complete file needs to be available before the video can start to play. This is not a problem when the files are stored on the local device (e.g. a microSD card) but if the files are to be downloaded over an Internet link then the index will need to be positioned at the start of the file so that playing the video can commence before the full file is downloaded³.

Setting up the user device

As stated previously, the system will work on a range of devices – but the user may need some advice on how to set up their specific device. The first appendix describes how to set up the system on an Android device. It is expected that instructions for other devices will be added later. Please get in contact with the authors if further advice is required.

² One organization that works with Deaf people has a group designing more than 100 icons for use with the main chronological Bible stories they translate and video.

³ This might be achieved with the software used to create the video, otherwise separate software can be obtained to do this. Further details on this point are beyond the scope of this documentation

Introduction to technical Concepts

The system is designed to be as simple to set up as is possible, to enable it to be applied by someone with only limited technical skills. With only one exception⁴ no technical knowledge is assumed. It has been designed such that by following a simple set of instruction a non-technical person can replicate, develop and maintain the system.

It is recommended that anyone developing this system should not create the files from scratch, but rather edit an existing file from the demonstration set. This will save time and ensure that the correct coding is used.

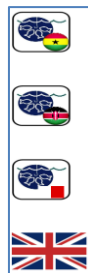
The system uses basic HTML concepts – but does not require people modifying the system to understand any HTML. They just need to be able to modify files as described in this document using a plain text editor⁵.

They also need to be able to build a hierarchical folder system which is both logical and extensible. (This last point is crucial if the system is to be easily manageable into the future by different people).

Visual Design concept

The user should be able to see a set of icons which are self-explanatory. Clicking on an individual icon will either take the user to the next set of icons to choose from, or play a video.

For example, consider the following opening screen of icons, which are contained in **folder level 0**:



Start screen (Level 0)

This will show the reader that contents are available in the following sign languages:

- Ghana Sign Language
- Kenyan Sign Language
- Kerala Sign Language
- British Sign Language (or/and Makaton)

By clicking the appropriate flag the reader will be taken to the start page for that language.

⁴ The only exception is the css file. It is not envisaged that the users will modify this file – but if difficulties with the display of the icons etc. are encountered, then they should consult with a web design specialist to amend this file.

⁵ For example notepad, although Notepad++ is a better editor (and can be downloaded for free from <http://notepad-plus-plus.org/>)

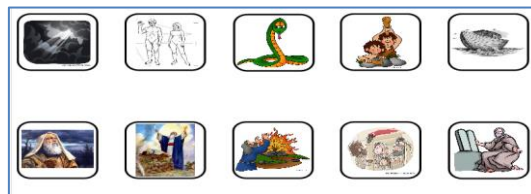
Clicking on the Ghana flag will bring up the following screen of icons, which are contained in a **folder at level 1**. For this particular set (Ghana Sign Language – GSL) there is only one folder at this level. Its main components are three folders, four icons, a video file and an HTML file.



Passage Group selection (Level 1)

Clicking the first icon will play a video giving a timeline for the whole Bible.

Clicking on the second icon will bring up the following screen of icons, which are contained in a **folder at level 2**. For this particular set there are 3 such folders, each holding 10-12 icons and an HTML file. This folder contains 10 folders, 10 icons and an HTML file.



Passage selection

The ten icons each represent a particular passage (in this case a biblical story). The icons have been drawn to visualise the contents.

Clicking on an icon, for example the snake, brings up a screen of icons similar to the following, which are contained in a **folder at level 3**. For this particular set, there are 32 such folders at this level and each folder contains four video files and an HTML file. The four icons are accessed from earlier folder levels as determined by the HTML file.



Passage Content (level 2)

Clicking on one of the icons will then play one of four videos as follows:

- Topic
- Introduction
- The story
- Conclusion

This sequence of clicking on icons and drilling down is typical of how the whole system works and is replicated throughout the system.

It is intuitive to the user, simple, fast and can be extended indefinitely.

Tools required to create the system

It is best to create the system on a PC and all that is required is a text editor⁵ and a browser.

Once completed and tested it can then be made available on other devices (see [Distribution](#) section below).

Implementing the design

We will now consider how to build the above example. It would be very helpful if you can download the level 0 folder and all sub-folders from the SLISS website so that you can check what is important.

The first step is to create a folder hierarchy as follows:

Level 0 – the root folder (in this example it contained icons for several different sign languages)

Level 1 – a set of 3 folders (under level 0) containing the passage group selection

Level 2 – 3 sets of folders (each set of 10-12 folders is in the appropriate level 1 folder) containing the passages within the group

Level 3 – in each of the 32 folders that are implemented at Level 2, there are instructions for activating a set of 4 videos - topic, introduction, story & conclusion.

The following is the arrangement of folders, icons & videos used for the demo. Only the Ghana Sign Language (GSL) set is fully implemented. It is the DOOR “God know how?” set. Look at the demo folder (and its sub-folders) and see how they are implemented:

Level-0 folder contains 6 folders and 5 icons for the languages	1 GSL level-1 folder containing 3 folders, 4 icons & 1 video	3 GSL level-2 folders. Each contains 10-12 folders & icons	32 GSL Level-3 folders. Each contains 4 videos & uses icons from other folders
Ghana (GSL)	a	10 folders & icons	
	b	10 folders & icons	
	c	12 folders & icons	
Kenya	Not implemented		
Kerala	Not implemented		
UK	BSL	Not implemented	
	Makaton	Not implemented	
USA	Not implemented		
Common images			

The next step is put the files in the folders. All folders are essentially the same with the exception of level 0 which must contain the CSS file⁶.

Each folder will contain one HTML file and several icon files (preferably PNG files, although JPEG files are also acceptable). The deepest level folders (Level 3 in this case) will each contain four video files paired with an icon. Upper level folders will (usually) contain a folder paired with each icon.

⁶ The contents of the css file are beyond the scope of this document. A developer can use the one provided in the sample set.

The HTML file is used to display the icons and navigate to the folder/video selected by the user. All HTML files follow essentially the same format. A sample system is available to illustrate – and the reader is strongly encouraged to look at the system to understand how the system is applied.

Let us take a look at the level 1 HTML file (index.htm) that is used for Ghana Sign Language (with line numbers added to aid discussion):

```
1  <!DOCTYPE >
2  <html>
3  <head>
4  <meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
5  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-
  scale=1.0, user-scalable=0," />
6  <meta name="HandheldFriendly" content="True" />
7  <title>Sign Language Project: Ghana Welcome Page</title>
8  <link href="../.styles.css" rel="stylesheet" type="text/css" />
9  </head>
10 <body>
11 <p>
12 <a href="Time Line.mp4"></a>
13 <a href="a/index.htm"></a>
14 <a href="b/index.htm"></a>
15 <a href="c/index.htm"></a>
16 </p>
17 </body>
18 </html>
```

This “index.htm” file will include a link to one video and three folders

Lines 1-6: should be included in all index files without alteration

Line 7: The text between <title> and </title> should be altered to describe the contents of the current folder

Line 8: This is required – but must be edited carefully. The href="../.styles.css" must be altered according to the level.

level 0 should be href=".styles.css"

level 1 should be href="../.styles.css"

level 2 should be href="../../.styles.css"

level 3 should be href="../../../.styles.css"

level 4 should be href="../../../../.styles.css"

Etc.

Lines 9-11: should be included without alteration

Lines 12-15: each line refers to an icon and what happens when the user clicks on the icon. The icons will appear in the order in which they are included in this file.

Careful examination of the first line will reveal that it is a link to a video using a consistent pattern as follows:

```
<a href="videofilename"></a>
```

You should include one complete line for each icon and replace *videofilename* and *iconfilename* with the actual file names (including file extension). You should also replace *description* with brief indication of what the link refers to.

Careful examination of the next three lines will reveal that they are links to sub-folders. Again a consistent pattern is used as follows:


```
<a href="foldername/index.htm"></a>
```

You should include one complete line for each icon and replace *foldername* with the name of the folder being linked to and the *iconfilename* with the actual icon file name (including the file extension). You should also replace *description* with brief indication of the contents of the new folder, i.e. what the link refers to.

[Note: At level 1 in this example there was only a need for 4 lines in this group, i.e. lines 12-15, but if there are 10 icons at a particular level, then this group would expand to lines 12-21 and following lines would be consequently re-numbered.]

Lines 16-18: should be included without alteration

Listening to audio files and displaying text files

To watch a video file the following line in the index.htm file was used:

```
<a href="videofilename"></a>
```

The concept behind of SLISS should also work with audio files and also for displaying text.

At the time of writing this documentation this has not been tested.

Please either send us an email to register your interest or (better) subscribe to our blog (<http://sliss.info>) to be kept informed of developments regarding use of audio and text files

Important note on filenames

Case sensitivity

It is important to remember when developing a system on a Windows computer that file names are not case sensitive on Windows. However on some other computers, tablets, phones etc. file names are case sensitive.

In practice this means that a) consistency of file names with exact case matching should always be used and b) if the system is created / updated using a Windows system then it should be checked / tested on a non-Windows based system.

Self-explanatory names

Whilst the computer is oblivious to the actual names used it is good practice to use self-explanatory names or names which conform to a known file reference system are used to ease maintenance of the system. This has been done in the examples below.

Files come in pairs, an icon file and an associated video file. Whilst not a technical requirement it is good practice to use the same name for the pair and differentiate the two by the file extension, e.g.

aStory.png and aStory.mp4

There is no requirement for the names to be in any particular sequence (the sequence in which icons are displayed is determined by their order in the html file and not the file name).

Hiding folders and files

The normal user does not want to see the files and the file structure on their computer. This is achieved on all Unix / Linux based systems (e.g. Apple and Android) by ensuring that all files and folder names in the root (with the **exception** of index.htm in the root) begin with a dot/period.

All windows based systems should have the hidden file attribute set for these files

Distribution

Before circulating the system on an SD Card, memory stick or equivalent they should be physically locked (where possible) to prevent accidental over-riding or deletion of contents.

Video formats

The system being described here relies on the browser or some other app on the device being used to play the videos⁷.

Unfortunately, at the time of writing, there is no single video format which is supported by all browsers. After some experimentation it has been found that videos created with the H.264 codec (MP4 format) are the most widely supported and it that is our current recommendation.

If the user's browser cannot play such video files direct (e.g. on a PC) then the user should use a different browser (e.g. Mozilla Firefox which is available for both PC's and Mac's).

If the user's browser starts a video player (e.g. on an Android phone) then the user should use a different video player (see footnote 10).

If the system is to be put on an internet connection then the video formats should have their index placed at the start of the file and not, as is common, at the end of the file. Otherwise the whole file must be downloaded before the video can start playing.

Icon size and formats

All icons should be drawn on a 630 x 630 canvas to produce consistency throughout the system. The only exception is the launcher icon which should be 144 x 144⁸

Icons should be saved in the png (Portable Network Graphics) format (although jpeg files are also acceptable, but not preferable).

⁷ The browser will either play the video directly (e.g. as on a PC) or a separate video player will be launched (e.g. as on an Android phone).



⁸ This is the launcher icon that was first used for a sign language implementation. By choosing a size of 144x144 pixels (the biggest size for an Android launcher icon) and using the Android "File Shortcut" app, the correct size icon was displayed on the home screen.

Frequently Asked Questions

How do we deal with two countries with the same sign language?

The first country icon should link to level 1 for that country. Subsequent countries should link to the level 1 for the first country – see how the USA and Canada are used in the sample file

How do we deal with a country which has more than one sign language?

This can be accommodated in one of two ways.

One way is to have modified level 0 icons which differentiate between the two languages. To illustrate see the following opening screen for the UK.



A second way requires introducing a new level 1 with icons which enable the user to select the particular sign language. All other levels (e.g. the passage selection level) are then moved up one level.

Use on Android phones / tablets

Android phones and tablets etc. have an app called “Gallery” which is used to show all the photos on the phone / tablet. When the system described here is stored on an Android phone / tablet the icons will also be included in the Gallery app. This is obviously not a good idea. To prevent this from happening an empty file named .NoMedia must be placed in each and every folder of this system.

Appendix 1: Setting up an Android Phone/Tablet for video files

Because of the open nature of Android devices there are many different ways to set up an Android device to work with the system. The following has been found to work, but the user may have a different setup on their device – as long as it works then that is fine.

Preparation

Initially prepare on a PC the desired folder structure, with the appropriate HTML files, icons and media files in the appropriate folders. When happy that clicking on the “index.htm” file in level 0 produces the expected icons and can play the expected media, copy the whole folder structure onto the SD card and insert it into the device.

The following apps must be installed on the device (those not already installed on your Android device can be downloaded and installed using the Android “Play Store” app):

- “**Default App Manager**” app
- An HTML viewer app: This can be a browser (**Chrome**⁹ has been proven to work well) or it can be the **HTML Viewer** that may already be installed on the device.
- For viewing videos, a video player app is required: **MX Player Pro**¹⁰ –works well, but has to be purchased by each user for his phone. There is a **free version**, but this includes adverts, which can show up when a video ends or is stopped.

[Note: If you use **MX Player** (the free version) as your video player, you can exit from a video at any point by using the Android back button, usually at the bottom of the smartphone. This means that the adverts are not much of an issue.]



- “**File Shortcut**” app.

Setting up the device

Ensure that the device automatically rotates content between portrait and landscape. To do this run the **settings** app

Settings → Display → Auto-rotate screen. Ensure that this is checked (ticked).

Set up some defaults by running the **Default App Manager**.

Clear any default app file associations for the browser and video player using the Settings app.

Default App Manager → Manage Apps → Browser

- Clear default
- Select the browser (Chrome) as your default browser using the ‘always’ option

Default App Manager → Video

⁹ Chrome is recommended as the browser since this is provided by Google who also provide the Android system. It is therefore specifically designed for the Android environment.

¹⁰ MX Player is a widely supported / used video player. The Pro version is recommended and the cost is small compared to the cost of the microSD card. But some users around the world would have no way to pay even this amount, and it is not difficult to deal with the adverts (see above). Alternatively, a developer could experiment with other free video players such as QQPlayer or MoboPlayer.

- Clear default
- Select MX Player Pro as the default video player using the 'always' option

Default App Manager → Video Streaming

- Clear default
- Select MX Player Pro as the default video player using the 'always' option

The device is now ready to be used.

Using the device

The simplest to set up but not best way to use the system is as follows:

1. Make sure that the SD card is already in the device.
2. Run the "File Manager" app and choose the SD Card
3. Navigate to folder that contains the SLISS set-up and click on the index.htm file.
4. Select the video you would like to view by clicking on the various icons which are displayed.

It would be better practice to be able to launch SLISS using an icon on the desktop.

1. The **simplest way** to do this is to navigate to the index.htm file via File Manager and then press the file name for a few seconds. A list of file options will then appear – one of which is to create a desktop shortcut for the file.
2. The **best way** is to create a desktop short cut icon with an appropriate icon – see footnote 8. One way to do this is to use a widget called "File Shortcut"
 1. Go to "Play Store" on your Android device or go to <https://play.google.com/store/apps/details?id=com.naqopy.file.shortcut&hl=en>.
 2. Download and install the "File Shortcut" app.
 3. Go to the home page and click on the icon that will allow you to see all the icons for the apps and widgets that have been installed. Touch "WIDGETS" at the top of the screen.
 4. Scroll, probably sideways, until you find the "File Shortcut" widget.
 5. Touch and hold the widget's icon and it will start.
 6. Touch the "Choose by FileOpenDialog" button
 7. Select the SD card (It might be sdcard, sdcard1, etc.) and navigate to the correct folder.
 8. Navigate to the index.html file (it will be below all files that begin with '.') – and then tap to select the file.
 9. Next touch the "Short Cut Name" field and enter a name of your choice for the system. **Do not press OK**, when you have finished this, but click on the down arrow to get rid of the keyboard.
 10. Scroll to the bottom of the screen and choose the icon you want, using "Choose from Gallery (auto resize)" and "File Manager" (the icon used in 8 is included on the sample system). If the icon does not show in the "File Shortcut" screen, choose another.
 11. Tap on OK