SCALP User Manual

ENAMEL Team:

Sunjik Lee

Li Yin

Bill Tzerpos

Table of Contents

1. Introduction 2  
1.1 How to Play 2  
1.2 How to Set Up 3

2. SCALP Program Functions 4  
2.1 Starting SCALP 4  
2.2 Entering Input 4  
2.3 Different Modes 4  
2.4 Settings 6

1. Introduction:

The Select-(as per)-Config-and-Launch-Player (SCALP) application is a suite of files that work together to create a cohesive function. It allows the user to select a scenario file for the program to play, as well as change the settings of SCALP, using buttons or keyboard keys.

* 1. Structure:

It consists of the following files, with file serves a particular function:

1. SCALP.sh
2. changeconfig.sh
3. startSCALP.sh
4. unmountDrive.sh
5. enamel.desktop
6. enamel\_usb\_autostart.rules
7. config.txt
8. The Java program, SCALP.java

SCALP.sh examines config.txt, which contain configuration variables, and acts accordingly to both config.txt and the current state of the Raspberry Pi. The current state can be either: the system has just booted up, or a USB flash drive was inserted, or the system has booted up with a USB flash drive inserted.

enamel.desktop simply runs SCALP.sh at bootup.

enamel\_usb\_autostart.rules runs SCALP.sh when a USB flash drive is inserted, or runs unmountDrive.sh when a USB flash drive is removed.

SCALP.java enables the user to interact with the SCALP program, giving audio feedback and allowing the user to provide input using hardware buttons (connected by Raspberry Pi’s GPIO pins), or through keyboard keys if the current system the program is being run on is not a Raspberry Pi. SCALP.java also includes an option within to change the variables in the config.txt file by using changeconfig.sh.

1.1 How to Set Up:

Go to the github link at (link) and download all the files under the Raspbian Scripts folder, either directly on the Raspberry Pi or on your computer and copy over to the Pi with a USB drive.

Place the following files in the **/home/pi/** directory on your Pi:

SCALP.sh, startSCALP.sh, unmountDrive.sh, changeconfig.sh, config.txt

Place the following file in the **/home/pi/.config/autostart/** directory:

enamel.desktop

Place the following file in the **/etc/udev/rules.d/** directory:

enamel\_usb\_autostart.rules

Run the following commands in the **terminal** application (built-in for Raspberry Pi):

sudo apt-get install oracle-java8-jdk

sudo apt-get install at

sudo update-alternatives

For this last command, select the appropriate java8 jdk you installed with the first command.

Go to the github link at (link) and download the files under Enamel. These files should be in a folder called Enamel; make sure **Enamel.jar** is in this folder, as well as the **FactoryScenarios** and **USBBuffer** directories. Your folder structure should look like this:

/home/pi/Enamel/FactoryScenarios/

/home/pi/Enamel/USBBuffer/

/home/pi/Enamel/mbrola/

/home/pi/Enamel/Enamel.jar

and all the other relevant .jar libraries, placed in /home/pi/Enamel/.

Once everything is set up, reboot your device, and it will now be ready for SCALP!

2. SCALP Program Functions

2.1 Starting SCALP

The SCALP Java program is started in two ways:

On device boot

On insertion of USB flash drive

If an instance of SCALP is already running, another start of SCALP will shut down the previous instance. For example, if you were in the middle of a scenario and you decide you want to play another file stored on a flash drive, *simply inserting that flash drive will restart SCALP automatically*, allowing you to select the scenario file in the flash drive.

Note that a blank Java program will open, a window with no components inside. This is normal; this is to allow the program to listen to keyboard key presses. As such, the Java window must be in focus for the keyboard keys to work. This is done by default, but if you switch focus to another window, just note that it will not work.

2.2 Entering Input

If you are using SCALP on the ENAMEL device, the program will recognize it and allow you to use the physical buttons on your device. Otherwise, the keyboard row’s keys can be used instead, which again will need the blank Java program window open and in focus. The program speaks out loud the instructions. Depending on the current menu, the buttons will behave differently.

2.3 The different modes of SCALP

SCALP has several modes, which depend on the current state of the system, as well as the current settings (see 2.4). The *default behaviour* is as follows:

* **If there is no USB flash drives inserted,** SCALP will only look at files on the device. The files it looks for are in two directories: FactoryScenarios and USBBuffer.
  + If there are no files in the USBBuffer directory, SCALP will ignore the “high level selector” and simply begin the program with the FactoryScenarios directory. This directory contains scenario files that are pre-loaded onto the device. You can cycle through the scenario files contained in FactoryScenarios using the buttons, and select which one to begin playing.
  + If there are files in the USBBuffer directory, SCALP will start in “high level selector”. In this mode, the first option presented is selecting between FactoryScenarios or USBBuffer. Once the option is selected, again you can cycle through the scenario files in that directory.
* **If there is a USB flash drive inserted,** then SCALP will only cycle through the files contained on the USB flash drive. If you do not want to play any of these scenarios, you can unplug the USB flash drive and the program will restart.
  + If the flash drive contains exactly one scenario, SCALP will select that scenario automatically and offer only a confirmation option to begin playing it.
  + If there are more than one scenarios on the flash drive, SCALP will copy the files over into the USBBuffer directory, and then start in “high level selector”. This is for convenience; once you hear the instructions for “high level selector”, this indicates that the files have finished copying over and you can remove the flash drive at any point. You do not have to keep the drive plugged in the whole time!
  + The scenarios on the flash drive must be kept in its root directory (i.e. not inside folders), and the audio files for the scenario must also be kept in a folder that’s in the flash drive’s root directory. *If any other, non-scenario files are on the flash drive, as long as they are not in.txt format*, it is not required to remove the non-scenario files before plugging it into the device.

*Note: By default, the device will only store 10 scenarios in USBBuffer. This is to make it manageable, and not fill up the microSD card the Raspberry Pi runs from. This is changeable in settings. The* ***oldest*** *scenario file is deleted if the number of files goes over 10.*

2.4 Settings

SCALP has a configuration file called “config.txt” that contains the default behaviour variables. Through the SCALP Java program, at any point in the program you may press the third button to enter settings. Again, the voice will guide you through the options. The settings that are changeable are as follows:

* Turn off/on auto-playing from the flash drive when there’s only one scenario file.
* Turn off/on auto-copying from the flash drive when there’s more than one scenario file,
* Changing the USBBuffer’s size (between 5, 10, 15, 20), meaning you can choose to store more or less scenario files.
* Turn off/on “smart clobber”. Smart clobber means that, when you plug in a flash drive, and the files on the drive and the files in USBBuffer are matching, those files will not be copied over. Comparison is done by date modified, meaning two scenarios (one in USBBuffer and one in the flash drive) with the same name but different content will not override the USBBuffer directory’s scenario.