

Guidelines for Developing Kindle Touch Homebrew Apps v0.1

These guidelines have been written in order to aid developers in creating apps by setting conventions and keeping files organized and to make the development and especially management of applications a lot easier.

1. The location of the files contained in an application

All apps should have their files stored in a single folder (if possible) located in the user partition. The path to the folder to any app should be `/mnt/us/HBApplications/application_id/`.

*HBApplications other possible names: Applications, Apps, Homebrew, etc

*application_id is described below

2. Files and folders contained in an application folder

a. Start.sh

This represents a uniform way to open applications. It can run other scripts, execute apps or open azw2 files.

b. A Settings folder

Located in `/mnt/us/extensions/application_name/Settings`, this folder should contain all the files that store the settings of the user. This way, applications will be easier to update and keep settings (and maybe backup settings).

c. An app_info.xml file containing the following information about the application:

- Application name
- Application ID (short name without special characters, usually the application's name written in CamelCase)
- Application version
- Developer name
- Application type (will be described below)

The apps should be categorized in the following types:

- **Simple** – defines an application that has files only in its folder (excluding the Launcher folder). These apps can easily be uninstalled by deleting its folder and entries in the Launcher and more importantly the Kindle should be able to update to newer firmwares without deleting these apps.
- **Complex** – defines an application that modifies system files. This application cannot be uninstalled by simply deleting its folder and would most likely have to revert the modifications it did to the system. Kindle Software Updates are very likely to fail when using these apps.
- **Extension** – These apps do not have entries in the launcher and cannot be normally accessed through the Launcher (like the screensaver hack). They should have their own folder here just like regular apps in order to keep a record of their installation, to contain the uninstallation script and for uniformity.

- **Switch** – These type describes apps that use On/Off switches (like usbnetwork).

An example XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
<information>
    <name>Really Cool App</name>
    <version>2.4</version>
    <author>John Doe</author>
    <type>normal</type>
    <id>reallycoolapp</id>
</information>
</application>
```

- d. An uninstallation script, uninstall.sh

This script will uninstall the app, cleaning its folder and reverting system changes.

- e. An icon file

In case the Launcher will be upgraded in the future to support Icons.

3. Distribution and installation

Apps should be distributed in packed folders (.zip / .tar.gz). At installation, the folder should be unpacked in /mnt/us/extensions/application_name/. After that a script (install.sh, found in the app's folder) should be run. This script will copy the necessary files to the Launcher folder (/mnt/us/extensions/Launcher/) in order for the app to be visible in the Launcher and do other changes to the filesystem required for the app to run like modify settings or replace system files (not recommended unless necessary).

This also opens opportunities for a simple package manager in the future. ☺ The package manager doesn't have to be very complex at all. It should handle browsing for installer archives, unpacking them and launching the install.sh. And also it should show a list of all the applications for uninstallation purposes and run uninstall.sh.

4. Uninstallation

All applications should be easily uninstallable by running the uninstall.sh script. This script has to delete application files and undo other system changes.

Other considerations:

The Launcher could be updated and converted into an asw2 app capable of displaying icons in a grid pattern (like other mobile OSes), maintain a simple database of the current apps (no app folders to describe apps) and include the simple package manager I described before. Since this proposed Launcher will also manage the Apps, maybe it should have its name changed.