

# Linux Application Suite (LAS)

## Desktop Environment Developer short paper

by Felix Caffier (broozar@web.de) draft 2015-03-30

## Content

1. Application bundle structure
2. Locations, Users and User Groups
3. LAS tools
4. File manager additions
5. Program Launcher additions

# 1. Application bundles

Application bundles are folders containing all files belonging to one user-space application. They (currently) can have the extension **.apx** (GUI app) or **.cap** (CLI app). Inside, you will find (among many others) 3 files that DE developers need to read/parse: **info.xml** (multilanguage name, description, build numbers etc.), **icon.png** (bundle icon) and **file.png** (associated file icon for files that the bundle application can read).

icon.png and file.png are square image files between 32x32 and 256x256px in size, preferably 128x128px.

The relevant parts of info.xml are:

```
<?xml version="1.0"?>
<app>
  [...]
  <category>Web;Games;Fun;</category>
  <name>
    <en>My Application</en>
    <de>Mein Programm</de>
    <en>Mi Aplicación</en>
  </name>
  <description>
    <en>A great program.</en>
    <de>Ein tolles Programm.</de>
    <en>Una aplicación muy bien.</en>
  </description>
  [...]
</app>
```

<?xml: because we like standards

root <app></app>: node name can be anything, will be ignored - but must be present

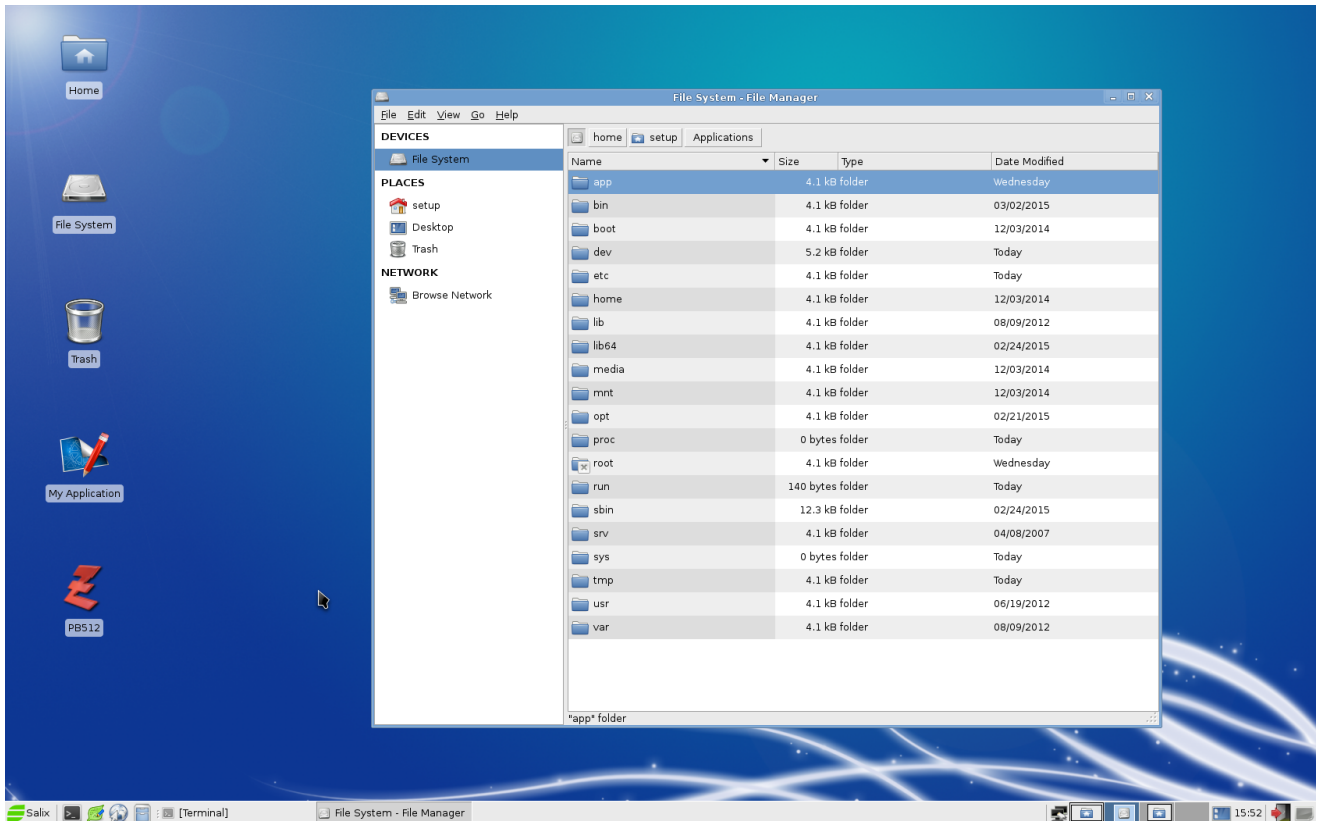
<category>: for launcher menus/desktop environments (optional)

<name> in <language>: program name, <en> is required, all other languages optional

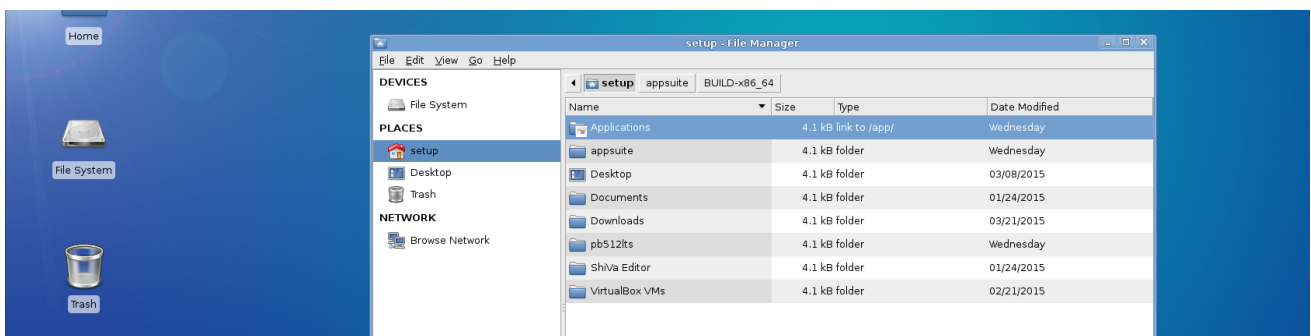
<description> in <language>: app description, <en> is required, all other languages optional

## 2. Locations, Users and User Groups

App bundles are supposed to be installed into a new folder called **/app/**, but can be placed and launched from anywhere, as long as there is an absolute path provided to the LAS tools (see section 3).



It is recommended to have a link to **/app/** in the user's **/home** in his own language:



**/app/** is owned by a new user called **appmin** and belongs to the group **apps**. Permissions for **/app/** are 751 (or 750). Users who wish to execute bundles must be added to the group "apps". By default, the bundle installed applies the permission set 751 to every file in the bundle.

**appmin** should be allowed to execute GUI applications in the current X session.

### 3. LAS tools

These are the tools and programs that ship or are required to work with LAS. First up, of course, the application packs themselves:

#### **/app/appname.apx(or .cap) - the root folder of the application**

apx indicates a GUI app, cap indicates a terminal/CLI app. They are structured equally, the only difference: The cap bundle launcher opens a terminal first, then executes launch-[arch].sh. This gets rid of the common CLI app steps “open terminal - cd to directory - type script interpreter name and file name to execute”.

... and now the tools:

#### **/usr/bin/appopen**

Opens .cap and .apx bundles

Usage: appopen /path/to/bundle.cap(or .apx) other parameters here

#### **/usr/bin/appinfo**

Reads info.xml, README, LICENSE etc. files from the app bundle; depends on xdg-open

Usage: appopen /path/to/bundle.cap(or .apx)

#### **/usr/bin/appdesktopshortcut**

Creates an appname.desktop file of a given .cap/.apx on /home/username/Desktop/

Usage: appdesktopshortcut /path/to/bundle.cap(or .apx)

#### **/usr/bin/messenger**

General purpose messaging window, can be used to display GUI messages from shell scripts.

#### **/usr/bin/st**

Simple terminal by <http://st.suckless.org/>, included as lightweight and DE-independent solution for CLI-based applications.

#### **/usr/bin/libmakegeneric**

Creates generic name links for .so files.

#### **xdg-open**

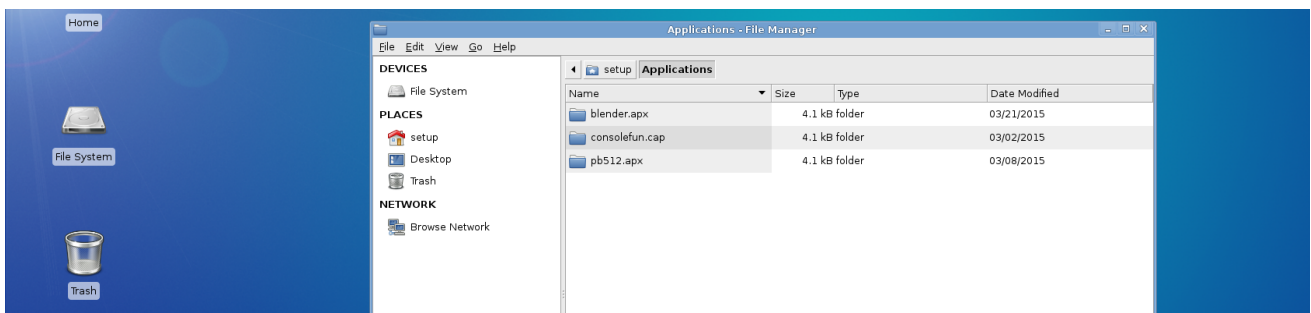
Dependency for appinfo and appinstall, not included in the LAS build.

## 4. File Manager additions

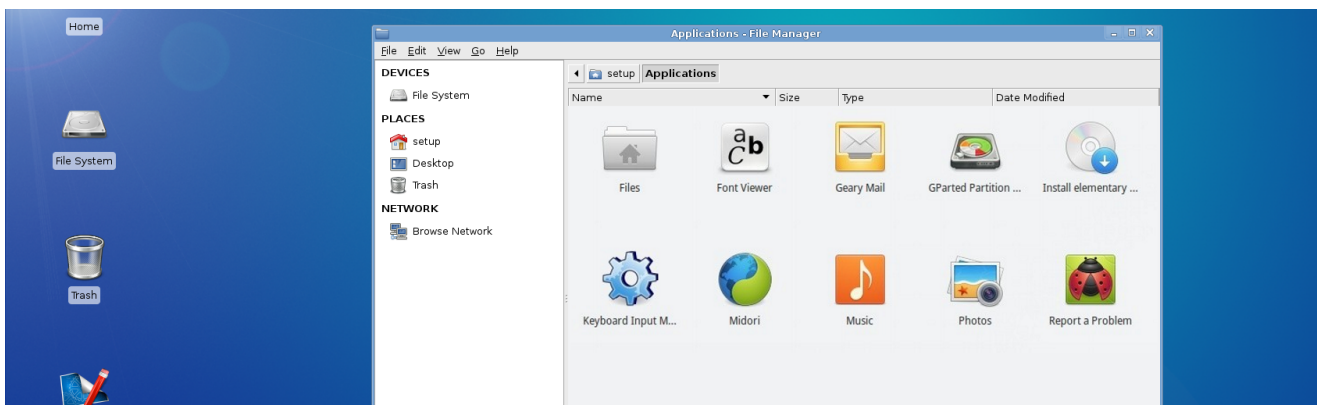
Whenever your File Manager encounters a folder with the .cap or .apx extension (ext),

1. read and parse *bundlename.ext/info.xml*, especially the app name in the proper language
2. read *bundlename.ext/icon.png* and replace the folder icon with icon.png
3. execute “*appopen /path/to/bundlename.ext*” if user double-clicks on icon
4. modify the context menu if user right-clicks on icon:
  - Show Contents (opens the bundle folder in the file manager)
  - Get Info (launches “*appinfo /app/bundlename.ext*”)
  - Create Desktop Shortcut (launches “*appdesktopshortcut /app/bundlename.ext*”)
  - Remove (launches some variant of *gksudo* with an “*rm*” command)

Visual mockup for (2): a generic menu like this...



... becomes this:



## 5. Program Launcher additions

The `/app/` folder needs to be searched and incorporated into your application launcher menu. Categories are supplied in the `info.xml` file. If you have a “fancy” launcher, you could also have a right-click menu there as described in section 4. I quite like the Deepin launcher:

