

Application entry points

# <sup>1</sup> Contents

2	Requirements	2
3	Security and privacy considerations	3
4	Menu entries	3
5	Background Services	3
6	Non-requirements	4
7	Recommendation	4
8	App identification	4
9	Desktop entries	4
10	Simple applications (one entry point)	5
11	Entry points which do not appear in the menus	5
12	Multiple-view applications	6

## **13** Requirements

17

18

27

28

35

<sup>14</sup> Flatpak application bundles<sup>1</sup> may contain *application entry points*, which are <sup>15</sup> any of these things:

- a graphical program<sup>2</sup> that would normally appear in a menu
  - a graphical program that would not normally appear in a menu, but can be launched in some other way, for example as a content-type handler  $^3$

Desktop environments provide metadata about these programs so that they can
be launched.

- <sup>21</sup> At least the following use-cases exist:
- It must be possible to translate the name into multiple languages, with a
   default (international English) name used for languages where there is no
   specific translation.
- Different manufacturers'launcher implementations might have a different taxonomy of categories for programs.
  - Certain graphical programs should be hidden from the menu, but treated as a first-class program during user interaction.

Some graphical programs present multiple views which may appear separately in menus, but are all implemented in terms of the same running process. For example, an audio player may appear in the menu three times, as "Albums", "Artists" and "Songs". However, ideally there would only be one audio player process at any given time, even if the user switches between views.

- Some programs should be started during device startup or user login.
- In the SDK images, Apertis applications and services should not necessarily be listed in the XFCE menus, and XFCE applications should not be

<sup>38</sup> listed in the "device simulator".

 $<sup>^{1}</sup> https://www.apertis.org/glossary/\#application-bundles$ 

 $<sup>^{2}</sup>$ https://www.apertis.org/glossary/#graphical-program

 $<sup>{}^{3}</sup> https://www.apertis.org/concepts/archive/application\_framework/content\_hand-over/$ 

#### Security and privacy considerations 39

The list of installed store application bundles is considered to be private infor-40 mation. for several reasons: 41

• the general operating principle for Apertis' app framework is that apps 42 must not affect each other, except where given permission to interact, 43 ensuring "loose coupling" between apps 44

• the presence of certain app bundles might be considered to be sensitive 45 (for example, app bundles correlated with political or religious views) 46

• the complete list could be used for user fingerprinting, for example guessing 47 that users of an online service share a device by observing that they have 48

the same list of app-bundles 49

The list of installed entry-points is almost equivalent to the list of store applica-50 tion bundles and has similar considerations. However, some components cannot 51 work without a list of store application bundles, or a list of their entry points. 52 This leads to some privacy requirements: 53

- Certain platform components such as the app launcher require the ability 54 to list store application bundles and/or their entry points. They must be 55 able to do so. 56
- Store applications with special permissions might also be allowed to list store application bundles and/or their entry points. 58
- Store applications may list the entry points that advertise a particular 59 *public interface*, as defined in the Interface discovery<sup>4</sup> design. 60
- Store applications without special permissions must not be able to enu-61 merate store application bundles that do not contain an entry point ad-62 vertising a public interface, either directly or by enumerating entry points 63 and inferring the existence of a bundle from its entry points. 64

#### Menu entries 65

57

Optionally, a single entry point may be specified to provide an icon for presen-66 tation in the application launcher. If no icon is presented it won't be obvious 67 to the user that they have the application installed, so the application store 68 screening process should carefully consider whether an application should be 69 allowed to install services and type handlers with no icon for the launcher. 70

#### **Background Services** 71

Background services in the current Flatpak distribution model do *not* use entry 72 points; rather, they request the ability to run arbitrary command lines in the 73 background at runtime via the XDG Background portal<sup>5</sup>. 74

<sup>&</sup>lt;sup>4</sup>https://www.apertis.org/concepts/archive/application\_framework/interface\_discovery/  ${}^{5}https://flatpak.github.io/xdg-desktop-portal/portal-docs.html \#gdbus-org.freedesktop.getablesktop.get$ portal.Background

## 75 Non-requirements

<sup>76</sup> System services<sup>6</sup> are outside the scope of this design.

## $_{77}$ Recommendation

Application bundle metadata<sup>7</sup> describes the fields that can appear in application
entry points and are expected to remain supported long-term. This document
provides rationale for those fields, suggested future directions, and details of
functionality that is not necessarily long-term stable.

### 82 App identification

Each built-in or store application bundle has a *bundle ID*, which is a reversed
 domain name<sup>8</sup> such as org.apertis.Example.

Each entry point within an application bundle has an *entry point ID*, which is
a reversed domain name such as org.apertis.Example.Entry.

For simple bundles with a single entry point, the bundle ID and the entry point
ID should be equal.

For more complex bundles with multiple entry points, the entry point ID should start with the bundle ID, but may have additional components.

All names should be allocated in a namespace controlled by the author of the bundle —in particular, Apertis applications should be in org.apertis. Sample code that is not intended to be used in production should be placed in com.example, with org.example and net.example also available for code samples that need to demonstrate the interaction between multiple namespaces (we prefer com.example, as a hint to developers that reversed domain names do not always start with "org").

## 98 Desktop entries

<sup>99</sup> Each Apertis *application entry point* is represented by a standard freedesk-<sup>100</sup> top.org Desktop Entry<sup>9</sup> (a .desktop file in XDG\_DATA\_DIRS/applications). The <sup>101</sup> desktop file must be named using the entry point ID, so org.apertis.Example <sup>102</sup> would have org.apertis.Example.desktop.

<sup>103</sup> The localestring<sup>10</sup> mechanism is used for translated strings.

Apertis applications should usually have OnlyShowIn=Apertis; so that they do not appear in the XFCE desktop environment menu in SDK images.

 $^{9} http://standards.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html \\$ 

<sup>&</sup>lt;sup>6</sup>https://www.apertis.org/glossary/#system-service

<sup>&</sup>lt;sup>7</sup>application-bundle-metadata.md

<sup>&</sup>lt;sup>8</sup>https://en.wikipedia.org/wiki/Reverse\_domain\_name\_notation

 $<sup>^{10}\</sup>rm https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.htm l#localized-keys$ 

<sup>106</sup> The Interfaces key is used for Interface discovery<sup>11</sup>. In particular, the following <sup>107</sup> interfaces are defined:

org.apertis.GlobalSearchProvider: Indicates that the application is a
 global search provider, equivalent to the supports-global-search schema
 entry.

The standard MimeType key controls the possible content-type and URI-scheme associations<sup>12</sup>. For example, x-scheme-handler/http is used in desktop environments such as GNOME to designate an application as capable of acting as a general-purpose web browser, and we will do the same here.

Services that parse desktop files should use the implementation in GLib<sup>13</sup>, or an Apertis-specific API built on top of that implementation.

Additional recommended keys The following additional keys are defined
in the [Desktop Entry] group.

x-GNOME-FullName (localestring): The human-readable full name of the application, such as Test Web Browser. This key is already used by the GLib library, and by desktop environments based on it (such as GNOME). Like Name, this is a "localestring": non-English versions can be provided with syntax like x-GNOME-FullName[pt\_BR]=Navegador Internet de teste.

<sup>124</sup> Simple applications (one entry point)

This is the simple case where an entry point has one "view", for example a web browser.

- 127 # cat org.test.Web.desktop
- 128 [Desktop Entry]
- 129 Type=Application
- 130 Name=Test Browser
- 131 GenericName=Browser
- 132 X-GNOME-FullName=The Test Web Browser
- 133 Exec=test-browser %U
- 134 Categories=Network;WebBrowser;
- 135 MimeType=text/html;x-scheme-handler/http;x-scheme-handler/https;
- 136 Icon=applications-internet

### 137 Entry points which do not appear in the menus

Some bundles might have an entry point that exists only to be started as a side-effect of other operations, for instance to handle URIs and content-types<sup>14</sup>.

<sup>&</sup>lt;sup>11</sup>https://www.apertis.org/concepts/archive/application\_framework/interface\_discovery/ <sup>12</sup>https://www.apertis.org/concepts/archive/application\_framework/content\_hand-over/

 $<sup>^{13} \</sup>rm https://gitlab.gnome.org/GNOME/glib/-/blob/main/gio/gdesktopappinfo.c\#L2007$ 

 $<sup>^{14}</sup> https://www.apertis.org/concepts/archive/application\_framework/content\_hand-over/application\_framework/content\_framework/conte$ 

Those entry points would have NoDisplay=true to hide them from the menus;
that is the only difference.

### 142 Multiple-view applications

Some bundles have more than one entry in the system menus; the example
referred to previously would be an audio player with entry points for Artists,
Songs, and Albums. We propose to represent these with one .desktop file per
menu entry.

In this model, each menu entry is a .desktop file. The audio player
would install org.test.Audio.Artists.desktop, org.test.Audio.Songs.desktop and
org.test.Audio.Albums.desktop. In addition, it would install org.test.Audio.desktop
with NoDisplay=true.

The running instance of the application would always identify itself as org.test.Audio, and the other three .desktop files would be linked to it by way of being in the same app bundle.

When using D-Bus activation<sup>15</sup> for applications (which is recommended), the application would have separate D-Bus .service files for all four names, would take all four bus names and their corresponding object paths at runtime, and would export the org.freedesktop.Application API at all four paths; but all of them would have SystemdService=org.test.Audio.service to ensure that only one activation occurs. The Activate, Open or ActivateAction method on each bus name would open the relevant view.

<sup>161</sup> The result would look something like this:

- 162 # org.test.Audio.desktop
- 163 [Desktop Entry]
- 164 Type=Application
- 165 Name=Frampton
- 166 GenericName=Audio Player
- 167 X-GNOME-FullName=The Test Audio Player
- 168 Exec=test-audio %F
- 169 Categories=Audio;Player;Music;
- 170 MimeType=audio/mpeg;
- 171 NoDisplay=true;
- 172 Icon=music
- 173 # org.test.Audio.Artists.desktop
- 174 [Desktop Entry]
- 175 Type=Application
- 176 Name=Frampton Artists
- 177 GenericName=Artists
- 178 Exec=test-audio --artists

 $^{15} \rm http://standards.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html \# dbus$ 

- 179 Categories=Audio;Player;Music;
- 180 Icon=music-artist
- 181 # org.test.Audio.Albums.desktop
- 182 [Desktop Entry]
- 183 Type=Application
- 184 Name=Frampton Albums
- 185 GenericName=Albums
- 186 Exec=test-audio --albums
- 187 Categories=Audio;Player;Music;
- 188 Icon=music-album
- 189 # org.test.Audio.Songs.desktop
- 190 [Desktop Entry]
- 191 Type=Application
- 192 Name=Audio Songs
- 193 GenericName=Songs
- 194 Exec=test-audio --songs
- 195 Categories=Audio;Player;Music;
- 196 Icon=music-track