



Apertis Application Bundle Specification

Contents

2	Introduction	3
3	Bundle ID	4
4	Reversed domain name	4
5	Top-level directory	5
6	Bundle metadata	5
7	Extended bundle metadata	7
8	AppArmor profile	8
9	Entry points	9
10	General fields for all entry points	10
11	Entry point ID	12
12	Main entry point	12
13	Content type and URI scheme handlers	12
14	Graphical programs	13
15	Multiple views	14
16	D-Bus activation	14
17	Agents	16
18	Paths for other file types	16
19	Executables	16
20	Libraries	17
21	Icon for the bundle	18
22	Icons for entry points	18
23	Icons for use by the bundle	19
24	Theme data for use by the bundle	19
25	GSettings schemas	19
26	Localized strings	20
27	Generic resource data	21
28	Example	22
29	Future directions	24
30	Appendix: built-in application bundles	24
31	Structural differences	24
32	Permissions and policy differences	25
33	Graphical programs	26
34	Command line arguments	26

This section describes the bundle specification for the [Canterbury legacy application framework](#)¹, which has been removed in v2022dev2 and is therefore not recommended for new development. See the [application framework](#)² document for further information on the new, replacement framework.

Version: 1.2.0 The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”

¹<https://www.apertis.org/architecture/canterbury-legacy-application-framework/>

²<https://www.apertis.org/concepts/application-framework/>

41 , “MAY”, and “OPTIONAL” in this document are to be interpreted as described
42 in [RFC 2119](#)³.

43 This specification uses [semantic versioning](#)⁴. After version 1.0.0 is finalized,
44 the major version number (first component) will be incremented if a change
45 makes previously-valid application bundles cease to work or be valid, for exam-
46 ple changing “MAY” to “MUST” or “MUST NOT”. The minor version number
47 (second component) will be incremented if a change makes previously-invalid
48 application bundles valid, for example changing “MUST” or “MUST NOT” to
49 “MAY”. The micro version number (third component) will be incremented for
50 editorial changes that do not affect the validity of an application bundle.

51 Introduction

52 This document aims to provide a stable filesystem layout for Apertis [store ap-
53 plication bundles](#)⁵ that can remain valid across multiple versions.

54 To keep older application bundles installable on newer Apertis releases, we an-
55 ticipate that incompatible changes (incrementing the major version) are to be
56 made very infrequently. If necessary, Apertis framework components might be
57 made to support multiple major versions of this specification.

58 A secondary goal of this specification is to provide a basis for the structure of
59 [built-in application bundles](#)⁶. Authors of built-in application bundles do not
60 necessarily need to limit themselves to the baseline set by this specification,
61 since a built-in application bundle will only be upgraded or rolled back at the
62 same time as a corresponding upgrade or rollback for the Apertis platform. How-
63 ever, by following the requirements and recommendations in this specification, a
64 built-in application bundle author can minimize the changes necessary between
65 Apertis platform releases. Please see Appendix: built in application bundles⁷
66 for differences between the required structure of store bundles and the required
67 structure of built-in application bundles.

68 App-bundles contain some or all of the following files:

- 69 • `bin/*` ([Executables](#))
- 70 • `etc/apparmor.d/Applications.bundle-ID` ([AppArmor profile](#))
- 71 • `lib/*` ([Libraries](#))
- 72 • `libexec/*` ([Executables](#))
- 73 • `share/applications/entry-point.desktop` ([Entry points](#))
- 74 • `share/glib-2.0/schemas/schema-ID.gschema.xml` ([GSettings schemas](#))
- 75 • `share/glib-2.0/schemas/gschemas.compiled` ([GSettings schemas](#))
- 76 • `share/icons/hicolor/64x64/apps/bundle-ID.png` ([Icon for the bundle](#))

³<https://datatracker.ietf.org/doc/html/rfc2119>

⁴<http://semver.org/>

⁵<https://www.apertis.org/concepts/applications/#software-categories>

⁶<https://www.apertis.org/concepts/applications/#software-categories>

⁷

- 77 • share/icons/hicolor/64x64/apps/*entry-point*.png (Icons for entry points)
- 78 • share/icons/*theme*/ (Icons for use by the bundle)
- 79 • share/locale/*/LC_MESSAGES/*.mo (Localized strings)
- 80 • share/metainfo/*bundle-ID*.appdata.xml (Bundle metadata)
- 81 • share/metainfo/*bundle-ID*.metainfo.xml (Bundle metadata)
- 82 • share/themes/*theme*/* (Theme data for use by the bundle)
- 83 • share/* (Generic resource data)

84 all of which will be installed relative to `/Applications/bundle-ID`.

85 To reduce its length, this specification does not generally provide rationale for
 86 its requirements. Please see the [Apertis concept designs](#)⁸ for design rationale,
 87 in particular the [Applications concept design](#)⁹, [Application Layout concept de-](#)
 88 [sign](#)¹⁰ and [Application Entry Points concept design](#)¹¹.

89 Bundle ID

90 Each Apertis application bundle has a *bundle ID*, which MUST consist of two or
 91 more components separated by dots (U+002E FULL STOP). Each component
 92 MUST start with an ASCII letter (A-Z, a-z) or underscore `_`, and contain only
 93 ASCII letters, underscores and ASCII decimal digits (0-9). Bundle IDs MUST
 94 NOT contain non-ASCII characters, for example accented letters such as `ä`.

95 Note that these are the same as the requirements for a [D-Bus inter-](#)
 96 [face name](#)¹², which are stricter than the requirements for a D-Bus
 97 bus name or a [GApplication application ID](#)¹³: every bundle ID is
 98 a valid bus name and a valid GApplication application ID, but not
 99 every bus name or application ID is a valid bundle ID.

100 Reversed domain name

101 The author of an application MUST choose a bundle ID that starts with a
 102 [reversed domain name](#)¹⁴ controlled by that author, with any hyphen/minus
 103 signs `-` replaced by underscores `_`, and `_` prepended to any component that
 104 starts with a digit.

105 For example, the owner of the domain name `collabora.com` controls the re-
 106 versed domain name `com.collabora` and might choose to name an app bundle
 107 `com.collabora.ShoppingList`.

108 Domain names with hyphen/minus signs, or with components starting with a
 109 digit, require special treatment to avoid syntactically invalid bundle IDs. If

⁸<https://designs.apertis.org/>

⁹<https://www.apertis.org/concepts/applications/>

¹⁰<https://www.apertis.org/concepts/application-layout/>

¹¹<https://www.apertis.org/concepts/application-entry-points/>

¹²<https://dbus.freedesktop.org/doc/dbus-specification.html#message-protocol-names>

¹³<https://developer.gnome.org/gio/stable/GApplication.html#g-application-id-is-valid>

¹⁴https://en.wikipedia.org/wiki/Reverse_domain_name_notation

110 the owner of 7-zip.org wishes to base bundle IDs on that domain name, they
111 MUST use a bundle ID starting with org._7_zip; for example, they might choose
112 to name an app bundle org._7_zip.Archiver.

113 Top-level directory

114 Each application bundle is made available on the user's system as a subdirectory
115 of /Applications whose name is the same as the bundle ID. App bundles MUST
116 NOT include any file outside that directory.

117 For example, the app bundle com.example.ShoppingList would use a top level
118 directory /Applications/com.example.ShoppingList/.

119 For brevity, this document will refer to this directory as \${prefix}.

120 Bundle metadata

121 Each app-bundle MUST install exactly one file in the \${prefix}/share/metainfo/
122 directory. The contents of that file are interpreted according to the [AppStream](#)
123 [upstream XML](#)¹⁵ specification.

124 This table provides a summary of the relevant tags. All other tags are either
125 not recommended for any type of bundle, or not allowed.

Tag	Status
id	required, must be bundle ID
name	required
summary	recommended
description	recommended
developer_name	recommended
metadata_license	required, should be CC0-1.0
project_license	optional
url	optional
releases	required
releases → release	required, exactly one
provides	optional
provides → dbus	optional
provides → <i>any other</i>	not allowed
custom → value	optional

126 If the app-bundle has **Entry points**, the file MUST be named either \${bun-
127 dle_id}.appdata.xml or \${bundle_id}.metainfo.xml, replacing \${bundle_id} with
128 the **Bundle ID**. In this case the component tag MUST have its type attribute set
129 to desktop.

¹⁵<https://www.freedesktop.org/software/appstream/docs/chap-Metadata.html>

130 If the app-bundle does not have **Entry points**, the file MUST be named `${bundle_id}.metainfo.xml`, again replacing `${bundle_id}` with the **Bundle ID**. In this
131 case the `component` tag MUST NOT have a `type` attribute.
132

133 The `id` tag MUST contain exactly the Apertis **Bundle ID**.

134 The `name` tag MUST contain a human-readable name for the app-bundle, for
135 example `Shopping List`.

136 The `summary`, `description` and `developer_name` tags SHOULD be present, with the
137 contents described by the [AppStream upstream XML](#)¹⁶ specification.

138 The `metadata_license` tag MUST be present, and MUST contain the identifier of
139 a permissive license under which the metadata can be redistributed. This license
140 SHOULD be the [Creative Commons Zero license](#)¹⁷, `CC0-1.0`, allowing unlimited
141 redistribution of the metadata with or without modifications (for example in
142 the user interface of an app-store).

143 The `metadata_license` does not imply anything about the terms under which
144 the app-bundle itself can be distributed: app-bundles themselves MAY be dis-
145 tributed under any license of their copyright holder's choice, including pro-
146 prietary licenses. The bundle metadata MAY represent that license in the
147 `project_license` tag, as described in the [AppStream upstream XML](#)¹⁸ specifi-
148 cation.

149 The `url` tag MAY be present, with the types and contents described by the
150 [AppStream upstream XML](#)¹⁹ specification.

151 The `releases` tag MUST be present, and MUST contain exactly one `release`
152 tag. The `release` tag MUST have a `version` attribute. Its value MUST start
153 with a digit and contain only digits and U+002E FULL STOP characters. Note
154 that this is a more strict requirement than in the AppStream upstream XML
155 specification, which allows additional `release` tags describing older releases.

156 Future directions:

157 This is a very strict versioning syntax, matching what Ribchester
158 accepts in Apertis 16.09. We should consider expanding this in a fu-
159 ture minor version of this specification to be able to accept dpkg-style
160 versions like `1.2.3~beta1+bugfix2`. This will require a formal specifi-
161 cation for how these version numbers are to be compared, possibly
162 deferring to [Debian Policy](#)²⁰.

163 The [Applications concept design](#)²¹ calls for the version number to
164 be split into an *application version* and a *store version*, analogous

¹⁶<https://www.freedesktop.org/software/appstream/docs/chap-Metadata.html>

¹⁷<https://creativecommons.org/publicdomain/zero/1.0/>

¹⁸<https://www.freedesktop.org/software/appstream/docs/chap-Metadata.html>

¹⁹<https://www.freedesktop.org/software/appstream/docs/chap-Metadata.html>

²⁰<https://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version>

²¹<https://www.apertis.org/concepts/applications/>

165 to the roles of the *upstream version* and *Debian revision* in Debian.

166 **Open question:** is the store version encoded in the `release` tag, or
167 is it stored in a `custom` tag or separately?

168 The `provides` tag MAY be present. It MAY contain a `dbus` tag, with its `type`
169 attribute set to `user`, for each well-known name provided by an entry point in
170 this application bundle. It MUST NOT contain any of the other child tags that
171 can be provided.

172 The `mimetypes` tag MUST NOT be present. In Apertis, content-type support is
173 handled by **Entry points**.

174 The `project_group` tag MUST NOT be present.

175 Tags that are not specified in the AppStream upstream XML specification
176 MUST NOT be present, with the exception of `custom` (see **Extended bundle**
177 **metadata**, below).

178 Tags not specified in this document, in particular `screenshots`, `suggests`, `trans-`
179 `lation` and `update_contact`, SHOULD NOT be present.

180 Extended bundle metadata

181 The bundle metadata MAY include one `custom` tag at the next level of hierarchy
182 below `component`. This tag MAY contain `value` child tags, each with a `key` at-
183 tribute and XML character data (text) content. It MUST NOT contain other
184 child tags or text.

185 Later versions of this specification will define keys starting with `x-Apertis-`. Keys
186 with that prefix that are not defined in this document MUST NOT be present.
187 The current version of this document does not define any such keys.

188 Other vendors MAY define keys starting with `x-` followed by a name distinctive
189 to the vendor.

190 Other keys SHOULD NOT be present.

191 For example, if a future version of this specification defined a key `x-`
192 `Apertis-ExampleColour`, and a vendor Wayne Industries defined a key `x-Wayne-`
193 `BatmobileCompatible`, this might result in bundle metadata like this:

```
194 <?xml version="1.0" encoding="UTF-8"?>
195 <component type="desktop">
196   <id>net.example.Extended</id>
197   <custom>
198     <value key="X-Apertis-ExampleColour">#00cc00</value>
199     <value key="X-WayneIndustries-BatmobileCompatible">true</value>
200   </custom>
201   ... additional metadata here ...
202 </component>
```

203 It is anticipated that this mechanism will be used for Apertis-specific
204 or automotive-specific extensions that are considered insufficiently
205 general to be included in the AppStream standard.

206 AppArmor profile

207 Apertis uses [AppArmor](#)²² to provide [security between application bundles](#)²³.
208 Each app-bundle MUST install exactly one AppArmor profile file at `${pre-
209 fix}/etc/apparmor.d/Applications.${bundle_id}`, replacing `${bundle_id}` with the
210 **Bundle ID**.

211 This file MUST define exactly one AppArmor profile. Its name MUST be exactly
212 `/Applications/${bundle_id}/**`, again replacing `${bundle_id}` with the bundle ID.
213 It MUST NOT have any [local profiles](#)²⁴ (also known as child profiles or subpro-
214 files), and in particular MUST NOT have any [hats](#)²⁵ (which are a special case
215 of local profiles).

216 This file SHOULD contain the following rules, replacing `@BUNDLE_ID@` with the
217 **Bundle ID** throughout:

```
218 /Applications/@BUNDLE_ID@/** {  
219     #include <abstractions/chaiwala-base>  
220     #include <abstractions/dbus-session-strict>  
221     #include <abstractions/fonts>  
222  
223     /Applications/@BUNDLE_ID@/{bin,libexec}/* pix,  
224     /Applications/@BUNDLE_ID@/{bin,lib,libexec}/{,**} mr,  
225     /Applications/@BUNDLE_ID@/share/{,**} r,  
226  
227     owner /var/Applications/@BUNDLE_ID@/users/** rwk,  
228  
229     owner link  
230         subset /var/Applications/@BUNDLE_ID@/users/**  
231         -> /var/Applications/@BUNDLE_ID@/users/**,  
232  
233     dbus send  
234         bus=session  
235         path=/org/freedesktop/DBus  
236         interface=org.freedesktop.DBus  
237         member={RequestName,ReleaseName}  
238         peer=(name=org.freedesktop.DBus),  
239     dbus bind bus=session name="@BUNDLE_ID@",
```

²²<https://gitlab.com/apparmor/apparmor/wikis/About>

²³<https://www.apertis.org/concepts/security/#security-between-applications>

²⁴[https://gitlab.com/apparmor/apparmor/wikis/AppArmor_Core_Policy_Reference#lo-
cal-profiles-and-hats](https://gitlab.com/apparmor/apparmor/wikis/AppArmor_Core_Policy_Reference#local-profiles-and-hats)

²⁵[https://gitlab.com/apparmor/apparmor/wikis/AppArmor_Core_Policy_Reference#lo-
cal-profiles-and-hats](https://gitlab.com/apparmor/apparmor/wikis/AppArmor_Core_Policy_Reference#local-profiles-and-hats)


```

240     dbus bind bus=session name="@BUNDLE_ID@.*",
241     dbus (send, receive) bus=session peer=(label=/Applications/@BUNDLE_ID@/**),
242     dbus receive bus=session peer=(label=/usr/bin/canterbury),
243
244     signal receive peer=/usr/bin/canterbury,
245 }

```

246 The profile MAY add additional permissions. The app-store curator is expected
247 to check additional permissions carefully.

248 Future direction: [the profile should be generated from simpler meta-](#)
249 [data](#)²⁶ in a future minor version of this specification.

250 Future direction: if we recommend particular interpreters—for exam-
251 ple `/bin/sh` for wrappers that set environment variables, a JavaScript
252 or Python interpreter for interpreted app code, or a webapp runtime
253 for HTML5 apps—then the generic profile recommendation should
254 allow those interpreters to be used.

255 Entry points

256 Each app-bundle MAY install `.desktop` files in the `${prefix}/share/applications/`
257 directory. The contents of that file are interpreted according to the [Desktop](#)
258 [Entry Specification](#)²⁷.

259 App bundles are not required to install any entry points at all, but many fea-
260 tures can only be provided by an app bundle that has entry points: **Graphical**
261 **programs** in the main menu MUST have an entry point, and **Content type and**
262 **uri scheme handlers** MUST have a **Main entry point**.

263 The name of each desktop entry file, excluding the `.desktop` extension, is called
264 the **Entry point ID**.

265 This table provides a summary of the allowed, recommended and optional fields.
266 All other fields are either not recommended for any type of entry point, or not
267 allowed. A table cell containing a literal value indicates that the field is required
268 and must have exactly that value.

Field	Main entry point	Other graphical programs	Agents
Categories	required	required	not recommended
Exec	required	required	required
GenericName	optional	optional	optional
Icon	required	required	not recommended
Interfaces	optional	optional	optional
MimeType	optional	not allowed	not allowed
Name	recommended	recommended	recommended

²⁶<https://phabricator.apertis.org/T311>

²⁷<http://standards.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html>

Field	Main entry point	Other graphical programs	Agents
NoDisplay	optional	optional	true
OnlyShowIn	Apertis;	Apertis;	Apertis;
Path	optional	optional	optional
Type	Application	Application	Application
X-Apertis-CategoryIcon	required	required	not recommended
X-Apertis-CategoryLabel	required	required	not recommended
X-Apertis-Type	application	application	agent-service
X-GNOME-FullName	optional	optional	optional
DBusActivatable	true recommended	true recommended	true recommended
X-Apertis-ServiceExec	recommended	optional	not allowed
X-Apertis-ParentEntry	not recommended	optional	not allowed

General fields for all entry points

Type MUST be set to Application.

OnlyShowIn MUST be set to Apertis;.

Exec MUST be present. The first word in Exec MUST be the absolute path to an executable in either `${prefix}/bin` or `${prefix}/libexec`.

Subsequent words in Exec MUST NOT use the % placeholders such as %F.

The Canterbury application manager does not support those placeholders.

Subsequent words in Exec MUST NOT be exactly `app-name`, `play-mode` or `url`, and SHOULD NOT be exactly `menu-entry`.

These words cause unexpected special behaviour in Apertis 16.06. After this special behaviour has been removed, future minor versions of this specification should remove this limitation.

Name SHOULD be specified. Its value MAY be a “brand name” such as `Firefox`, a generic name such as `Web browser`, or a combination of the two such as `Firefox web browser`.

GenericName SHOULD be specified if its value would differ from Name. If present, its value MUST be a generic (unbranded) name such as `Web browser`, for use in user interfaces whose designer wishes to standardize on generic names.

X-GNOME-FullName SHOULD be specified if its value would differ from Name. If present, its value SHOULD be a full name incorporating both the brand name and the generic name, for example `Firefox web browser`, suitable for use in situations where it is necessary to disambiguate between entry points with the same GenericName (for example if both Firefox and Chrome have `GenericName=Web browser`).

294 Translated versions of these names, such as `Name[fr]`, MAY be present using the
 295 [localestring](#)²⁸ mechanism defined in the Desktop Entry Specification.

296 `Path` MAY be set, with its usual meaning (it sets the current working directory
 297 for the program). If `Path` is not set, programs in the app-bundle will inherit the
 298 working directory of the parent process, and MUST NOT assume that it will
 299 take any particular value.

300 `Interfaces` MAY be set, for [interface discovery](#)²⁹.

301 `DBusActivatable` MAY be present and set to `true`, as described in [D-Bus activa-](#)
 302 [tion](#).

303 `X-Apertis-ServiceExec` MAY be set, as described in [D-Bus activation](#).

304 `X-Apertis-ParentEntry` MAY be set, as described in [Multiple views](#).

305 The following keys MUST NOT be present:

- 306 • `Encoding` (the encoding MUST be UTF-8, which is the default)
- 307 • `Hidden` (this misleadingly named key is used to mark entry points as
 308 deleted, which is not useful in this context)
- 309 • `NotShowIn`
- 310 • `StartupNotify`
- 311 • `StartupWMClass`
- 312 • `Terminal`
- 313 • `URL`
- 314 • `Version` (version 1.0 of the Desktop Entry specification is assumed)

315 The following keys SHOULD NOT be present, and application bundles
 316 SHOULD NOT rely on their normal functionality (if any):

- 317 • `Actions`
- 318 • `Comment`
- 319 • `Environment`
- 320 • `Keywords`
- 321 • `TryExec`
- 322 • `X-Apertis-AudioChannelName`
- 323 • `X-Apertis-AudioResourceOwner`
- 324 • `X-Apertis-AudioRole`
- 325 • `X-Apertis-BackgroundState`
- 326 • `X-Apertis-BandwidthPriority`
- 327 • `X-Apertis-DataExchangeRules` (obsolete)
- 328 • `X-Apertis-ManifestUrl` (obsolete)
- 329 • `X-Apertis-SettingsIcon` (set an `Icon` instead)
- 330 • `X-Apertis-SettingsName` (set the name in the [Bundle metadata](#) instead)

²⁸<https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#localized-keys>

²⁹https://www.apertis.org/concepts/interface_discovery/

- X-Apertis-SettingsPath (use the mechanism described in [GSettings schemas](#) instead)
- X-Apertis-SplashScreen
- X-Apertis-WindowName (obsolete)

Future directions:

X-Apertis-AudioRole, X-Apertis-BackgroundState and X-Apertis-BandwidthPriority are under consideration for a future minor version of this specification, but are not currently considered to be stable.

Entry point ID

Each entry point MUST have an *entry point ID*, which is a string with the same syntax requirements as a **Bundle ID**. The name of the `.desktop` file MUST be the entry point ID followed by `.desktop`.

Like the **Bundle ID**, all entry point IDs in an app-bundle MUST start with a **Reversed domain name** controlled by the author. It is RECOMMENDED that all entry point IDs in an app-bundle either match its bundle ID exactly, or start with the bundle ID followed by a dot.

A single executable program MAY be represented by more than one entry point.

If a program will request a D-Bus well-known bus name to provide interfaces to graphical programs in the same bundle, the well-known bus name MUST be the same as one of its entry point IDs.

Main entry point

Each app-bundle that has any entry points SHOULD have an entry point whose **Entry point ID** is exactly the **Bundle ID**. This entry point is referred to as the *main entry point*, and MUST be a [graphical program][Graphical programs].

Certain metadata fields of the main entry point, including its `Categories`, `Icon` and `MimeType`, are copied into the cache of bundle metadata during installation and hence made available to platform applications.

Content type and URI scheme handlers

The **Main entry point** MAY be registered as the **content type handler**³⁰ for media types such as `audio/mpeg`, by setting `MimeType` to a list of **content types**³¹, each followed by a semicolon `;`. Non-main entry points MUST NOT be content type handlers. For example, a media player with support for the MP3 and RealAudio formats might use `MimeType=audio/mpeg;audio/vnd.rn-realaudio;` in its entry point.

³⁰https://www.apertis.org/concepts/content_hand-over/

³¹<https://www.iana.org/assignments/media-types/media-types.xhtml>

365 The **Main entry point** MAY be registered as the handler for **URI schemes**³² such
366 as `tel` or `http`, by including an entry in `MimeType` for the pseudo-content-type `x-`
367 `scheme-handler/scheme`, for example `x-scheme-handler/http` for a web browser.
368 Non-main entry points MUST NOT be URI scheme handlers.

369 If the entry point implements **D-Bus activation**, sending the `org.freedesktop.Application.Open`
370 `method call`³³ to the object path corresponding to its entry point ID MUST
371 result in it attempting to open the URI or URIs passed as parameters.

372 Graphical programs

373 Each graphical program (user interface, HMI) that will be directly launched
374 by the user MUST have an entry point. Each graphical program that will be
375 associated with content types or URIs MUST have an entry point.

376 A graphical program MAY have more than one entry point, to appear in menus
377 more than once (for example, the Frampton media player uses this³⁴ to appear
378 three times under the names Artists, Albums and Songs).

379 Graphical programs MUST set `X-Apertis-Type` to `application`.

380 This is required by Apertis 16.09, but might be phased out in a later
381 minor version of this specification.

382 If a graphical program is intended to be shown in the menus, `NoDisplay` MUST
383 NOT be specified. Otherwise, it MUST be specified and set to `true` (for exam-
384 ple, the Frampton media player uses this to allow its *main entry point* to be
385 associated with media file types while hiding it from the menus).

386 `Categories` MUST be set to a list of appropriate menu categories from the
387 freedesktop.org [Desktop Menu Specification](https://standards.freedesktop.org/menu-spec/latest/apa.html)³⁵, each followed by a semicolon
388 `;`. There MUST be at least one Main Category.

389 `X-Apertis-CategoryLabel` MUST be set to the human-readable English name of
390 a single category, which MUST be in title-case with no special formatting (for
391 example, `Video & TV` is correct, while `V I D E O & T V` is not).

392 This is required by Apertis 16.09, but should be phased out in favour
393 of having launchers parse `Categories` in a later minor version of this
394 specification.

395 `X-Apertis-CategoryIcon` MUST be set to the name of the icon to be used for the
396 category in launchers, with no `/` characters or file-type extension, for example
397 `icon_music_AC`. The icon MUST be chosen from among those provided by the
398 platform's launcher (the allowed values are therefore platform-specific).

³²<https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml>

³³[https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.htm](https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus)
[l#dbus](https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus)

³⁴<https://gitlab.apertis.org/appfw/frampton/tree/v0.6.1/scripts>

³⁵<http://standards.freedesktop.org/menu-spec/latest/apa.html>

399 This is required by Apertis 16.09, but should be phased out in favour
400 of having launchers parse Categories in a later minor version of this
401 specification.

402 `Icon` MUST be set to the name of either the **Icon for the bundle** or one of
403 the **Icons for entry points**, as a bare icon name (without any `/` characters, and
404 without a file-type extension such as `.png` or `.svg`). In particular, this implies
405 that its string value MUST match either the **Bundle ID**, or the **Entry point ID**
406 of an entry point.

407 Multiple views

408 Some application designs have a group of entry points that are all implemented
409 by invoking the same executable with different parameters, all implemented in
410 the same process. For example, a music player might have separate entry points
411 to view the music library grouped by artist or album, or as a single flat list of
412 songs.

413 In applications that work like this, one of these views MUST be nominated to
414 be the *parent entry point*, with the others as *child entry points*. The parent will
415 usually be the **Main entry point**, although this is not required. The main entry
416 point SHOULD NOT be a child entry point.

417 A parent entry point MUST NOT have the `X-Apertis-ParentEntry` field. It MUST
418 set `DBusActivatable` to `true`, and implement [D-Bus activation] for its own entry
419 point ID and the entry point IDs of all associated child entry points. It SHOULD
420 set `X-Apertis-ServiceExec`.

421 Child entry points MUST set the `X-Apertis-ParentEntry` field to the **Entry point**
422 **ID** of the parent entry point, and MUST set `DBusActivatable` to `true`. They
423 MUST NOT set `X-Apertis-ServiceExec`.

424 Agents and other non-graphical programs MUST NOT be parent or child entry
425 points.

426 D-Bus activation

427 Programs in an app-bundle MAY declare that they implement *D-Bus activa-*
428 *tion* by setting `DBusActivation` to `true` in each entry point that starts the same
429 program.

430 Graphical program entry points that set `DBusActivation` to `true` and do not have
431 an `X-Apertis-ParentEntry` field SHOULD also have an `X-Apertis-ServiceExec` field.
432 The `X-Apertis-ServiceExec` field has the same syntax as the standard `Exec` field.

433 Agents MUST NOT have an `X-Apertis-ServiceExec` field, since their `Exec` field
434 has essentially the same meaning.

435 We define the *service activation command line* to be the `X-Apertis-ServiceExec`
436 field if present, or the `Exec` field otherwise.

437 The service activation command line MUST be a command-line that will
 438 start the program without opening any graphical windows, such that it will
 439 be ready to receive D-Bus requests. If a program uses the [GApplication](#)³⁶
 440 API (which is recommended), then the service activation command line for
 441 graphical programs will typically be the absolute path of the executable
 442 followed by a space and the `--gapplication-service` argument, for example
 443 `X-Apertis-ServiceExec=/Applications/com.example.ShoppingList/bin/main --`
 444 `gapplication-service`, while the service activation command line for agents and
 445 other non-graphical programs (with the `G_APPLICATION_IS_SERVICE` flag³⁷) will
 446 typically just be the path to the executable.

447 When the service activation command line for a graphical program is launched,
 448 the resulting process MUST export a D-Bus object path that is derived from
 449 the entry point ID by prepending `/` and replacing each `.` with `/`, then re-
 450 quest a well-known name equal to the entry point ID. The interfaces of that
 451 object path MUST include at least the `org.freedesktop.Application` interface³⁸,
 452 and MAY include additional standard or non-standard interfaces such as the
 453 `org.gtk.Application` interface³⁹ used by GLib's [GApplication](#)⁴⁰ objects. When
 454 launched in this way, the process MUST NOT behave as though any of its en-
 455 try points were activated until it receives an appropriate D-Bus method call; in
 456 particular, it MUST NOT open any windows until it is told to do so.

457 When the service activation command line for an agent or non-graphical pro-
 458 gram is launched, the resulting process MAY export a D-Bus object path im-
 459 plementing `org.freedesktop.Application` as above, but is not required to do so.
 460 If it does, the `Activate` and `Open` methods are not required to be implemented,
 461 since they are unlikely to be useful for non-graphical programs.

462 If an entry point has an `X-Apertis-ParentEntry` field (a [child entry
 463 point][Multiple views]), when the parent entry point named in that field
 464 is started by its service activation command line, the resulting process MUST
 465 also export D-Bus object paths and request well-known names corresponding
 466 to the entry point IDs of each of its child entry points.

467 For graphical programs, sending the `org.freedesktop.Application.Activate`
 468 D-Bus method call⁴¹ to one of the object paths described above MUST
 469 result in the program displaying whatever window is appropriate for the
 470 corresponding entry point. If the graphical program implements [content-type
 471 handling][Content type and uri scheme handlers], then the same is true for the

³⁶<https://developer.gnome.org/gio/stable/GApplication.html>

³⁷<https://developer.gnome.org/gio/stable/GApplication.html#G-APPLICATION-IS-SERVICE:CAPS>

³⁸<https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus>

³⁹<https://wiki.gnome.org/Projects/GLib/GApplication/DBusAPI>

⁴⁰<https://developer.gnome.org/gio/stable/GApplication.html>

⁴¹<https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus>

472 [org.freedesktop.Application.Open method](#)⁴². This requirement is not applicable
473 to agents and other non-graphical programs.

474 The process MAY export additional object paths and interfaces. It SHOULD
475 NOT request additional well-known names.

476 When the `Exec` command of a D-Bus-activatable graphical entry point is
477 launched, the resulting process MUST arrange for a program to be run (directly
478 or indirectly) that will request the well-known name corresponding to that
479 entry point ID, export the corresponding D-Bus object path, and behave as
480 though that object path had received an `Activate` or `Open` method call, modified
481 according to the command-line arguments if appropriate: in other words, it has
482 behaviour similar to the `Exec` command of a non-D-Bus-activatable graphical
483 program. If the program uses the [GApplication](#)⁴³ API, this will normally
484 be achieved by setting `Exec` to the absolute path of the executable, with no
485 arguments, for example `Exec=/Applications/com.example.ShoppingList/bin/main`.
486 This requirement is not applicable to agents and other non-graphical programs.
487 See the [specification of the Application interface](#)⁴⁴ for more details about its
488 methods.

489 Agents

490 Each agent (background service) MUST have an entry point.

491 Agents MUST set `X-Apertis-Type` to `agent-service`.

492 This is required by Apertis 16.09, but might be phased out in a later
493 minor version of this specification.

494 Agents MUST set `NoDisplay` to `true`.

495 Agents SHOULD set `DBusActivatable` to `true`, and implement **D-Bus activation**
496 as described above.

497 Paths for other file types

498 Executables

499 Any executable programs in the app-bundle MUST be installed in either the
500 `${prefix}/bin` or `${prefix}/libexec` directory, or a descendant directory in `${pre-`
501 `fix}/libexec`. For example, these paths are valid:

502 `${prefix}/bin/my-executable`

503 `${prefix}/libexec/my-helper-executable`

504 `${prefix}/libexec/other-helper/other-helper-executable`

⁴²[https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.htm](https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus)
l#dbus

⁴³<https://developer.gnome.org/gio/stable/GApplication.html>

⁴⁴[https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.htm](https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus)
l#dbus

505 Suitable directories are conveniently available as `${bindir}`, `${libex-`
506 `ecdir}` and `${pkglibexecdir}`‘ when using Automake.

507 Libraries

508 An Apertis application bundle MAY contain private libraries for use by that
509 application bundle, for example shared libraries written in C or C++, or Python
510 modules.

511 If present, architecture-dependent library files MUST be located in the `${pre-`
512 `fix}/lib` directory or a descendant of that directory. Architecture-independent
513 library files such as “pure Python” modules MUST be located in either the `${pre-`
514 `fix}/lib` or `${prefix}/share` directory, or a descendant of one of those directories.

515 For example, the app bundle `com.example.ShoppingList` might contain library
516 files `/Applications/com.example.ShoppingList/lib/libwebapi.so.0` or `/Applica-`
517 `tions/com.example.ShoppingList/lib/python3/webapi/__init__.py`.

518 Native executables SHOULD be linked with a `DT_RPATH` pointing to the location
519 of their required libraries. For example, the `ShoppingList` app bundle described
520 above might be linked using `gcc -Wl,-rpath=/Applications/com.example.ShoppingList/lib`.

521 If the app bundle is built using GNU automake and libtool, this will
522 typically be done automatically.

523 Programs in app-bundles MUST NOT assume that any special environ-
524 ment variables to locate libraries, such as `LD_LIBRARY_PATH`, `GI_TYPELIB_PATH`
525 or `PYTHONPATH`, will be set by the application framework. For example, if
526 the `ShoppingList` app bundle described above needs to be able to load
527 `/Applications/com.example.ShoppingList/lib/python3/webapi/__init__.py` via
528 the Python statement `import webapi`, it cannot assume that `/Applica-`
529 `tions/com.example.ShoppingList/lib/python3` is already in `sys.path`. Its main
530 executable might prepend that directory to `sys.path`, or its main executable
531 might be a shell script that sets `PYTHONPATH` and then runs the underlying
532 Python code with `exec`.

533 A possible change in future minor versions of this specification would
534 be to set a specified list of environment variables used by a specified
535 set of recommended libraries, such as `LD_LIBRARY_PATH` for libc and
536 `GI_TYPELIB_PATH` for GObject-Introspection. Python is not among our
537 recommended frameworks, so we would probably still not include
538 `PYTHONPATH`.

539 For each native ELF library, the app-bundle MUST contain a file whose name
540 exactly matches the [SONAME](#)⁴⁵ (ELF `DT_SONAME`) of the library, in a directory
541 that will be searched by all executables that use that library (for example via
542 `DT_RPATH` or `LD_LIBRARY_PATH`). This file MUST either be a regular file (the library
543 itself), or a symbolic link to the library’s “real name”.

⁴⁵<http://tldp.org/HOWTO/Program-Library-HOWTO/shared-libraries.html>

544 Building and installing shared libraries using GNU libtool is RECOMMENDED:
545 libraries built like this will typically have a correct symbolic link from the SONAME
546 to the “real name” without further action from the developer.

547 For example, if the ShoppingList app-bundle has executables linked to a private
548 library whose SONAME is `libwebapi.so.0`, it might include a regular file with exactly
549 that name; or it might include a regular file named `libwebapi.so.0.1.2`, and a
550 symbolic link `libwebapi.so.0 → libwebapi.so.0.1.2`.

551 Icon for the bundle

552 The app-bundle MAY have an icon to represent the bundle as a whole, in a
553 generic user interface icon theme. The generic user interface icon theme is rep-
554 resented by the reserved theme name `hicolor`, as required by the freedesktop.org
555 [Icon Theme Specification](#)⁴⁶.

556 If the app-bundle has this icon, it MUST be in [Portable Network Graphics](#)⁴⁷
557 format, 64×64 pixels in size, and MUST be located at

558 `${prefix}/share/icons/hicolor/64x64/apps/${name}.png`

559 where `${name}` is set to the **Bundle ID**.

560 **Open question:** I’m arbitrarily choosing 64x64 because that’s what
561 the AppStream specification uses, but do we have a different pre-
562 ferred size in Apertis?

563 To minimize display artefacts caused by resizing, the app-bundle MAY make this
564 icon available in some or all of the additional sizes used in the freedesktop.org
565 [reference implementation](#)⁴⁸ of the `hicolor` fallback theme (8, 16, 22, 24, 32, 36,
566 42, 48, 64, 72, 96, 128, 192, 256 or 512 pixels). These MUST be installed to the
567 corresponding path with `64x64` replaced by the appropriate size.

568 The app-bundle MAY have variations of this icon that fit better in specific user
569 interface themes. If present, these MUST be installed to the corresponding
570 path with `hicolor` replaced by the name of the intended theme. For example, if
571 a theme named `net.example.Metallic` is popular, an app-bundle might include
572 a version of its own icon that has been designed to coordinate well with the
573 Metallic theme, at

574 `${prefix}/share/icons/net.example.Metallic/64x64/apps/${name}.png`

575 Icons for entry points

576 Any entry point MAY have an icon to represent it. If present, it MUST be
577 named in the same way as the icon for the bundle as a whole, except that
578 `${name}` is set to the **Entry point ID** instead of the bundle ID.

⁴⁶<https://specifications.freedesktop.org/icon-theme-spec/icon-theme-spec-latest.html>

⁴⁷<https://datatracker.ietf.org/doc/html/rfc2083>

⁴⁸<https://www.freedesktop.org/software/icon-theme/releases/>

579 Note that this means the **Main entry point** of the app-bundle will always use
580 the same icon as the bundle itself.

581 Icons for use by the bundle

582 The app-bundle MAY contain other icons. They SHOULD be arranged accord-
583 ing to the freedesktop.org **Icon Theme Specification**⁴⁹.

584 For example, if the app-bundle is an email client, it might include a `mail-mark-`
585 `important` icon for use by a “Mark as Important” button. If it has a generic
586 version for use by unrecognised themes, and that generic version is 24 pixels in
587 size, that version might be installed in:

```
588 ${prefix}/share/icons/hicolor/24x24/actions/mail-mark-important.png
```

589 If the app-bundle also has a version for use by a popular theme named
590 `net.example.Metallic`, it might install that as:

```
591 ${prefix}/share/icons/net.example.Metallic/24x24/actions/mail-mark-important.png
```

592 The app-bundle MAY assume that it will be launched with the `XDG_DATA_DIRS`
593 **environment variable**⁵⁰ set to a value that includes `${prefix}/share`, so that com-
594 mon icon theme implementations such as `GtkIconTheme`⁵¹ will automatically
595 use icons from the `${prefix}`.

596 Theme data for use by the bundle

597 The app-bundle MAY install theme data into subdirectories of `${pre-`
598 `fix}/share/themes` whose names correspond to theme names.

599 The app-bundle MAY assume that it will be launched with the `XDG_DATA_DIRS`
600 **environment variable**⁵² set to a value that includes `${prefix}/share`, so that
601 common theme implementations such as `GtkCssProvider`⁵³ will automatically
602 use theme data from the `${prefix}`.

603 GSettings schemas

604 **GSettings schemas**⁵⁴ are used for **preferences**⁵⁵.

605 The app-bundle MAY install one or more **GSettings schemas**⁵⁶ into
606 `${prefix}/share/glib-2.0/schemas/`. The filenames used MUST be the schema
607 ID followed by `.gschema.xml`, optionally accompanied by enum definitions in a
608 file named by the schema ID followed by `.enums.xml`. Each schema ID SHOULD

⁴⁹<https://specifications.freedesktop.org/icon-theme-spec/icon-theme-spec-latest.html>

⁵⁰<https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html>

⁵¹<https://developer.gnome.org/gtk3/stable/GtkIconTheme.html>

⁵²<https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html>

⁵³<https://developer.gnome.org/gtk3/stable/GtkCssProvider.html>

⁵⁴<https://developer.gnome.org/gio/stable/GSettings.html>

⁵⁵<https://www.apertis.org/concepts/preferences-and-persistence/#preferences-approach>

⁵⁶<https://developer.gnome.org/gio/stable/GSettings.html>

609 either match the **Bundle ID** exactly, or start with the bundle ID followed by a
610 dot.

611 If the app-bundle installs any schemas, then it **MUST** also install
612 a compiled binary form of those schemas, in `${prefix}/share/glib-`
613 `2.0/schemas/schemas.compiled`. The `glib-compile-schemas` tool can be used
614 to compile this binary form.

615 The app-bundle **MAY** install a schema whose schema ID matches the **Bundle**
616 **ID** exactly. If it does, then that schema's child schemas **MUST** all start with
617 the bundle ID followed by a dot, and that schema and its child schemas will be
618 made available in the system settings user interface.

619 If the app-bundle author does not intend for it to appear in the system set-
620 tings user interface, then the app-bundle **MUST NOT** use its bundle ID as a
621 schema ID. It **MAY** use an alternative schema ID such as `${bundle_id}.Internal`,
622 resulting in a schema file named `${bundle_id}.Internal.gschema.xml`.

623 The app-bundle **MAY** assume that it will be launched with the `XDG_DATA_DIRS`
624 **environment variable**⁵⁷ set to a value that includes `${prefix}/share`, so that
625 GSettings will automatically use these schemas.

626 Localized strings

627 Some file formats, such as `.desktop` files and AppStream XML, put **localized**
628 **strings**⁵⁸ in a single file, typically built from an international English version
629 and a set of translations at build-time. For the following file formats, the app-
630 bundle **MUST** include all of its supported translations (for example a translated
631 Name) in a single file:

- 632 • **Entry points**
- 633 • **Bundle metadata**

634 Otherwise, application bundles that contain localized strings **SHOULD** use **GNU**
635 **gettext**⁵⁹ `.mo` files. These **SHOULD** be stored in the `${prefix}/share/locale`
636 hierarchy, with a subdirectory named for the *locale* in which the language is
637 used, and a `LC_MESSAGES` subdirectory inside that containing one or more `.mo`
638 files. The name of the `.mo` files (the *text domain*) **SHOULD** either be exactly
639 the **Bundle ID**, or the bundle ID followed by a dot and one or more additional
640 components. Using a single text domain whose name is exactly the bundle ID
641 is **RECOMMENDED**.

642 For example, if the app bundle `com.example.ShoppingList` is localized into generic
643 international French, French as spoken in Canada, and Uzbek written in Cyrillic,
644 it might contain:

- 645 • `/Applications/com.example.ShoppingList/share/locale/fr/LC_MESSAGES/com.example.ShoppingList.mo`

⁵⁷<https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html>

⁵⁸<https://www.apertis.org/architecture/internationalization/>

⁵⁹<https://www.gnu.org/software/gettext/manual/index.html>

- /Applications/com.example.ShoppingList/share/locale/fr_CA/LC_MESSAGES/com.example.ShoppingList.mo
- /Applications/com.example.ShoppingList/share/locale/uz@cyrillic/LC_MESSAGES/com.example.ShoppingList.mo

The other `LC_` directories used by `gettext` MAY exist alongside `LC_MESSAGES`.

If using `gettext`, programs in the app bundle would typically have to make API calls similar to these to activate these localized strings:

```
setlocale (LC_ALL, "");
bindtextdomain (GETTEXT_PACKAGE, DATADIR "/locale");
bind_textdomain_codeset (GETTEXT_PACKAGE, "UTF-8");
textdomain (GETTEXT_PACKAGE);
```

where `DATADIR` would be defined to `\`${datadir}\`` by the build system (expanded to `/Applications/com.example.ShoppingList/share` at build time), and `GETTEXT_PACKAGE` would be defined to `com.example.ShoppingList` in this example.

In general, use of `gettext` is not mandatory, and neither is this specific layout. Application bundles MAY store localized strings in any format of their choice, in any subdirectory of `${prefix}/lib` or `${prefix}/share`. If this is done, the application bundle author is responsible for arranging for those localized strings to be loaded.

There is one special case where use of `gettext` and this specific layout *is* mandatory. If an app bundle contains **GSettings schemas**, and those schemas support localized contents by using the `gettext-domain` attribute, then the `gettext-domain` that is declared MUST be either the **Bundle ID**, or the bundle ID followed by a dot and one or more additional name components. Again, using exactly the bundle ID for the `gettext` domain is RECOMMENDED.

One possible direction for a future minor version would be to allow GSettings schemas to include inline translations, similar to `.desktop` files. This would require GLib modifications: at the moment this is specifically not allowed by GLib.

Generic resource data

Non-executable resource files such as graphics and sounds MUST be located in either the `${prefix}/lib` or `${prefix}/share` directory, or a descendant of one of those directories.

CPU-architecture-dependent resource files MUST be located in the `${prefix}/lib` directory or a descendant of that directory. CPU-architecture-independent resource files SHOULD be located in the `${prefix}/share` directory or a descendant of that directory.

The app-bundle MAY assume that it will be launched with the `XDG_DATA_DIRS` environment variable⁶⁰ set to a value that includes `${prefix}/share`, so that any library that uses that variable (for example via `g_get_system_data_dirs()`) will

⁶⁰<https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html>

684 automatically load resource files from the appropriate subdirectory of `${prefix}-`
685 `fix}/share`.

686 The app-bundle MUST NOT assume that the application framework will set
687 environment variables that make it load resource files from `${prefix}/lib`.

688 Example

689 For example, suppose the owner of `example.net` produces an application named
690 Shopping List, with a graphical program to display shopping lists, and a back-
691 ground agent to pop up reminders when the vehicle is driven near a supermarket.
692 Suppose the agent provides a D-Bus API to the graphical program.

693 Suppose this application also opens the `application/vnd.example.shoppinglist`
694 content type, and handles `myproduct:` URIs.

695 Suppose the bundle ID is `net.example.ShoppingList`, so the bundle's files will
696 be available at `/Applications/net.example.ShoppingList`. The minimal metadata
697 required for this bundle might resemble what is shown in this section; all paths
698 are given relative to `/Applications/net.example.ShoppingList`, which we will refer
699 to as `${prefix}`.

700 [Application bundle metadata][Bundle metadata], to be installed as `${pre-`
701 `fix}/share/metainfo/net.example.ShoppingList.appdata.xml`:

```
702 <?xml version="1.0" encoding="UTF-8"?>
703 <component type="desktop">
704   <id>net.example.ShoppingList</id>
705   <metadata_license>CC0-1.0</metadata_license>
706   <name>Shopping List</name>
707   <summary>Keep track of your groceries</summary>
708   <description>
709     <p>Never run out of cornflakes again with this easy-to-use shopping
710       list manager, featuring:</p>
711     <ul>
712       <li>Special offer notifications</li>
713       <li>Driving directions to the nearest supermarket</li>
714       <li>Cloud synchronization</li>
715     </ul>
716   <developer_name>Example Software Inc.</developer_name>
717   <url type="homepage">https://example.net/shopping-list</url>
718   <release version="1.0" date="2016-08-23" />
719 </component>
```

720 The [settings schema][GSettings schemas] would be installed to `${prefix}/share/glib-`
721 `2.0/schemas/net.example.ShoppingList.gschema.xml`, optionally accompanied by
722 `${prefix}/share/glib-2.0/schemas/net.example.ShoppingList.enums.xml`. Those
723 files would be compiled into `${prefix}/share/glib-2.0/schemas/gschemas.compiled`,

724 for example by using a command like `glib-compile-schemas --strict ${DEST-
725 DIR}${prefix}/share/glib-2.0/schemas` while building the bundle.

726 **Localized strings** used in the app itself, or in its GSettings schema, would be in-
727 stalled as `${prefix}/share/locale/${locale}/LC_MESSAGES/com.example.ShoppingList.mo`,
728 where `${locale}` represents a locale such as `fr_CA` or `de`.

729 **Main entry point** for the user interface, to be installed as `${prefix}/share/applications/net.example.ShoppingList`

```
730 [Desktop Entry]
731 Categories=Utility;
732 Exec=/Applications/net.example.ShoppingList/bin/gui
733 GenericName=Shopping List
734 Icon=net.example.ShoppingList
735 MimeType=application/vnd.example.shoppinglist;x-scheme-handler/myproduct;
736 Name=Shopping List
737 OnlyShowIn=Apertis;
738 Type=Application
739 X-Apertis-Type=application
740 X-GNOME-FullName=Example Shopping List
741 DBusActivatable=true
742 X-Apertis-ServiceExec=/Applications/net.example.ShoppingList/bin/gui      --
743 gapplication-service
```

744 The user interface's [icon][Icon for the bundle] would be installed as `${pre-
745 fix}/share/icons/hicolor/64x64/apps/net.example.ShoppingList.png`.

746 [Entry point][Entry points] for the agent, to be installed as `${pre-
747 fix}/share/applications/net.example.ShoppingList.Agent.desktop`:

```
748 [Desktop Entry]
749 Exec=/Applications/net.example.ShoppingList/bin/agent
750 NoDisplay=true
751 OnlyShowIn=Apertis;
752 Type=Application
753 X-Apertis-Type=agent-service
754 X-GNOME-FullName=Example Shopping List
755 DBusActivatable=true
```

756 **AppArmor profile**, to be installed as `${prefix}/etc/apparmor.d/Applications.net.example.ShoppingList:`

```
757 /Applications/net.example.ShoppingList/** {
758     #include <abstractions/chaiwala-base>
759     #include <abstractions/dbus-session-strict>
760
761     /Applications/net.example.ShoppingList/{bin,libexec}/* pix,
762     /Applications/net.example.ShoppingList/{bin,lib,libexec}/{,**} mr,
763     /Applications/net.example.ShoppingList/share/{,**} r,
764
765     owner /var/Applications/net.example.ShoppingList/users/** rwk,
```

```

766
767     dbus send
768         bus=session
769         path=/org/freedesktop/DBus
770         interface=org.freedesktop.DBus
771         member={RequestName,ReleaseName}
772         peer=(name=org.freedesktop.DBus),
773     dbus bind bus=session name="net.example.ShoppingList",
774     dbus bind bus=session name="net.example.ShoppingList.*",
775     dbus (send, receive) bus=session
776         peer=(label=/Applications/net.example.ShoppingList/**),
777     dbus receive bus=session peer=(label=/usr/bin/canterbury),
778
779     signal receive peer=/usr/bin/canterbury,
780 }

```

781 Future directions

782 Future versions of this specification could include layout and contents spec-
783 ifications for particular categories of [system extensions](#)⁶¹, in particular user-
784 installable UI themes and language packs.

785 Appendix: built-in application bundles

786 Built-in application bundles are maintained as part of the platform, and so are
787 outside the scope of this specification. However, their structure is similar.

788 As a general principle, built-in application bundles that closely resemble a store
789 application bundle, other than the structural differences listed here, will be as
790 portable between platform versions as a similar store application bundle would
791 be. Built-in application bundles that diverge more from that model will be more
792 tightly-coupled to the platform for which they were designed, and so are more
793 likely to need alterations for newer platform versions.

794 Structural differences

795 In general, [built-in application bundles](#)⁶² MUST have a structure analogous to
796 store application bundles, replacing `/Applications` with `/usr/Applications` in all
797 path prefixes. In particular, the `${prefix}` for a built-in application bundle is
798 `/usr/Applications/` followed by the bundle ID.

799 As an exception to the usual use of the `${prefix}`, built-in application bun-
800 dles MUST install their [AppArmor profiles][AppArmor profile] directly to
801 `/etc/apparmor.d`, in a file named `/etc/apparmor.d/usr.Applications.${bundle_id}`

⁶¹<https://www.apertis.org/concepts/applications/#system-extensions>

⁶²<https://www.apertis.org/concepts/applications/#software-categories>

802 where `${bundle_id}` is to be replaced by the **Bundle ID**. They MUST NOT
803 contain `/usr/Applications/*/etc/apparmor.d`.

804 For the following categories of files, if an equivalent store application bundle
805 would include files in that category, built-in application bundles MUST install
806 the real files into `${prefix}/share`. Additionally, the `.deb` file for the built-in
807 application bundle must include symbolic links `/usr/share/*` pointing to the
808 corresponding regular files in `${prefix}/share/*`:

- 809 • **Entry points**
- 810 • **GSettings schemas**
- 811 • **Icon for the bundle**
- 812 • **Icons for entry points**
- 813 • **Bundle metadata**

814 For example, the `.deb` file for a built-in application bundle `org.apertis.Eye` might
815 include a symbolic link `/usr/share/applications/org.apertis.Eye.desktop` point-
816 ing to the main entry point's real file `/usr/Applications/org.apertis.Eye/share/applications/org.apertis.Eye.desktop`
817 and similar symbolic links for GSettings schemas, icons and the bundle meta-
818 data.

819 For the following categories of files, if an equivalent store application bundle
820 would include files in that category, built-in application bundles MUST install
821 the files into `${prefix}`, but MUST NOT include symbolic links to them in
822 `/usr/*`:

- 823 • **Executables**
- 824 • **Libraries**
- 825 • **Icons for use by the bundle**
- 826 • **Localized strings**
- 827 • **Theme data for use by the bundle**
- 828 • **Generic resource data**

829 Permissions and policy differences

830 Recommendations and requirements that refer to the app-store curator do not
831 apply to built-in application bundles. The platform vendor has total control
832 over both the **platform layer**⁶³ and the built-in application bundles that are
833 packaged with it; they are responsible for ensuring that those components fit
834 together correctly and meet their functional and security requirements. For
835 example, a platform vendor can provide any **AppArmor profile** for a built-in
836 application bundle, and it is up to the platform vendor to ensure that the profile
837 is consistent with their security policy.

⁶³<https://www.apertis.org/concepts/applications/#software-categories>

838 Graphical programs

839 Built-in application bundles do not necessarily need to provide their own user
840 interfaces if they rely on an underlying service, for example one that is running
841 in the `automotive domain`⁶⁴, to display a user interface. Where this specification
842 calls for a particular entry point to be a graphical program, that requirement
843 or recommendation does not apply to built-in application bundles. A built-in
844 application bundle could provide similar functionality by communicating with
845 other processes, either locally or in the automotive domain, and arranging for
846 those other processes to display graphics instead.

847 However, if this is done, then the built-in application bundle is necessarily some-
848 what tightly coupled to the component to which it delegates its user interface.

849 Command line arguments

850 Built-in app-bundles SHOULD NOT use the `play-mode`, `app-name` or `url` tokens in
851 their `Exec` arguments. This is a weaker prohibition than for store app-bundles,
852 which MUST NOT use those tokens. This exception is made for backwards
853 compatibility. Please note that the special case made for these tokens in and
854 before Apertis 17.03 is deprecated, and their effect will change in future releases.

855 Agents and other non-graphical programs in built-in app-bundles SHOULD
856 NOT have an `X-Apertis-ServiceExec` field. This is a weaker prohibition than
857 for non-graphical programs in store app-bundles, which MUST NOT have that
858 field: it allows those agents and non-graphical programs to make use of the spe-
859 cial tokens like `play-mode` when run on Apertis 17.03, without including them in
860 the service command-line. This exception is made for backwards compatibility,
861 and is considered deprecated.

⁶⁴<https://www.apertis.org/concepts/inter-domain-communication/#automotive-domain>