



## Apertis Application Bundle Specification

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

35 **Version: 1.2.0**

36 The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL  
37 NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and  
38 “OPTIONAL” in this document are to be interpreted as described in RFC  
39 2119<sup>1</sup>.

40 This specification uses [semantic versioning](http://semver.org/)<sup>2</sup>. After version 1.0.0 is finalized,  
41 the major version number (first component) will be incremented if a change

<sup>1</sup><https://tools.ietf.org/html/rfc2119>

<sup>2</sup><http://semver.org/>

42 makes previously-valid application bundles cease to work or be valid, for example  
43 changing “MAY” to “MUST” or “MUST NOT”. The minor version number  
44 (second component) will be incremented if a change makes previously-invalid  
45 application bundles valid, for example changing “MUST” or “MUST NOT” to  
46 “MAY”. The micro version number (third component) will be incremented for  
47 editorial changes that do not affect the validity of an application bundle.

## 48 Introduction

49 This document aims to provide a stable filesystem layout for Apertis [store ap-](#)  
50 [plication bundles](#)<sup>3</sup> that can remain valid across multiple versions.

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

55 A secondary goal of this specification is to provide a basis for the structure of  
56 [built-in application bundles](#)<sup>4</sup>. Authors of built-in application bundles do not  
57 necessarily need to limit themselves to the baseline set by this specification,  
58 since a built-in application bundle will only be upgraded or rolled back at the  
59 same time as a corresponding upgrade or rollback for the Apertis platform. How-  
60 ever, by following the requirements and recommendations in this specification, a  
61 built-in application bundle author can minimize the changes necessary between  
62 Apertis platform releases. Please see Appendix: built in application bundles<sup>5</sup>  
63 for differences between the required structure of store bundles and the required  
64 structure of built-in application bundles.

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

- 66 • `bin/*` ( [Executables](#) )
- 67 • `etc/apparmor.d/Applications.bundle-ID` ( [AppArmor profile](#) )
- 68 • `lib/*` ( [Libraries](#) )
- 69 • `libexec/*` ( [Executables](#) )
- 70 • `share/applications/entry-point.desktop` ( [Entry points](#) )
- 71 • `share/glib-2.0/schemas/schema-ID.gschema.xml` ( [GSettings schemas](#) )
- 72 • `share/glib-2.0/schemas/gschemas.compiled` ( [GSettings schemas](#) )
- 73 • `share/icons/hicolor/64x64/apps/bundle-ID.png` ( [Icon for the bundle](#) )
- 74 • `share/icons/hicolor/64x64/apps/entry-point.png` ( [Icons for entry points](#) )
- 75 • `share/icons/theme/` ( [Icons for use by the bundle](#) )
- 76 • `share/locale/*/LC_MESSAGES/*.mo` ( [Localized strings](#) )
- 77 • `share/metainfo/bundle-ID.appdata.xml` ( [Bundle metadata](#) )
- 78 • `share/metainfo/bundle-ID.metainfo.xml` ( [Bundle metadata](#) )

---

<sup>3</sup><https://jwd.pages.apertis.org/apertis-website/concepts/applications/#software-categories>

<sup>4</sup><https://jwd.pages.apertis.org/apertis-website/concepts/applications/#software-categories>

<sup>5</sup>

- 79 • `share/themes/theme/*` ( Theme data for use by the bundle)
- 80 • `share/*` ( Generic resource data)

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

82 To reduce its length, this specification does not generally provide rationale for  
83 its requirements. Please see the [Apertis concept designs](#)<sup>6</sup> for design rationale, in  
84 particular the [Applications concept design](#)<sup>7</sup>, [Application Layout concept design](#)<sup>8</sup>  
85 and [Application Entry Points concept design](#)<sup>9</sup>.

## 86 Bundle ID

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

92 Note that these are the same as the requirements for a [D-Bus inter-](#)  
93 [face name](#)<sup>10</sup>, which are stricter than the requirements for a D-Bus  
94 bus name or a [GApplication application ID](#)<sup>11</sup>: every bundle ID is  
95 a valid bus name and a valid GApplication application ID, but not  
96 every bus name or application ID is a valid bundle ID.

## 97 Reversed domain name

98 The author of an application MUST choose a bundle ID that starts with a  
99 [reversed domain name](#)<sup>12</sup> controlled by that author, with any hyphen/minus  
100 signs `-` replaced by underscores `_`, and `_` prepended to any component that  
101 starts with a digit.

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

105 Domain names with hyphen/minus signs, or with components starting with a  
106 digit, require special treatment to avoid syntactically invalid bundle IDs. If  
107 the owner of `7-zip.org` wishes to base bundle IDs on that domain name, they  
108 MUST use a bundle ID starting with `org._7_zip`; for example, they might choose  
109 to name an app bundle `org._7_zip.Archiver`.

---

<sup>6</sup><https://designs.apertis.org/>

<sup>7</sup><https://jwd.pages.apertis.org/apertis-website/concepts/applications/>

<sup>8</sup><https://jwd.pages.apertis.org/apertis-website/concepts/application-layout/>

<sup>9</sup><https://jwd.pages.apertis.org/apertis-website/concepts/application-entry-points/>

<sup>10</sup><https://dbus.freedesktop.org/doc/dbus-specification.html#message-protocol-names>

<sup>11</sup><https://developer.gnome.org/gio/stable/GApplication.html#g-application-id-is-valid>

<sup>12</sup>[https://en.wikipedia.org/wiki/Reverse\\_domain\\_name\\_notation](https://en.wikipedia.org/wiki/Reverse_domain_name_notation)

## 110 **Top-level directory**

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

114 For example, the app bundle `com.example.ShoppingList` would use a top level  
115 directory `/Applications/com.example.ShoppingList/`.

116 For brevity, this document will refer to this directory as `${prefix}`.

## 117 **Bundle metadata**

118 Each app-bundle MUST install exactly one file in the `${prefix}/share/metainfo/`  
119 directory. The contents of that file are interpreted according to the [AppStream](#)  
120 [upstream XML](#)<sup>13</sup> specification.

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

Tag	Status
<code>id</code>	required, must be bundle ID
<code>name</code>	required
<code>summary</code>	recommended
<code>description</code>	recommended
<code>developer_name</code>	recommended
<code>metadata_license</code>	required, should be cc0-1.0
<code>project_license</code>	optional
<code>url</code>	optional
<code>releases</code>	required
<code>releases</code> → <code>release</code>	required, exactly one
<code>provides</code>	optional
<code>provides</code> → <code>dbus</code>	optional
<code>provides</code> → <i>any other</i>	not allowed
<code>custom</code> → <code>value</code>	optional

123 If the app-bundle has **Entry points**, the file MUST be named either  
124 `${bundle_id}.appdata.xml` OR `${bundle_id}.metainfo.xml`, replacing `${bundle_id}`  
125 with the **Bundle ID**. In this case the `component` tag MUST have its `type` attribute  
126 set to `desktop`.

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

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

<sup>13</sup><https://www.freedesktop.org/software/appstream/docs/chap-Metadata.html>

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

133 The `summary`, `description` and `developer_name` tags SHOULD be present, with the  
134 contents described by the [AppStream upstream XML](#)<sup>14</sup> specification.

135 The `metadata_license` tag MUST be present, and MUST contain the identifier of  
136 a permissive license under which the metadata can be redistributed. This license  
137 SHOULD be the [Creative Commons Zero license](#)<sup>15</sup>, `CC0-1.0`, allowing unlimited  
138 redistribution of the metadata with or without modifications (for example in  
139 the user interface of an app-store).

140 The `metadata_license` does not imply anything about the terms under which  
141 the app-bundle itself can be distributed: app-bundles themselves MAY be dis-  
142 tributed under any license of their copyright holder's choice, including pro-  
143 prietary licenses. The bundle metadata MAY represent that license in the  
144 `project_license` tag, as described in the [AppStream upstream XML](#)<sup>16</sup> speci-  
145 fication.

146 The `url` tag MAY be present, with the types and contents described by the  
147 [AppStream upstream XML](#)<sup>17</sup> specification.

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

153 Future directions:

154 This is a very strict versioning syntax, matching what Ribchester  
155 accepts in Apertis 16.09. We should consider expanding this in a  
156 future minor version of this specification to be able to accept dpkg-  
157 style versions like `1.2.3~beta1+bugfix2`. This will require a formal  
158 specification for how these version numbers are to be compared,  
159 possibly deferring to [Debian Policy](#)<sup>18</sup>.

160 The [Applications concept design](#)<sup>19</sup> calls for the version number to  
161 be split into an *application version* and a *store version*, analogous  
162 to the roles of the *upstream version* and *Debian revision* in Debian.

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

---

<sup>14</sup><https://www.freedesktop.org/software/appstream/docs/chap-Metadata.html>

<sup>15</sup><https://creativecommons.org/publicdomain/zero/1.0/>

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

<sup>17</sup><https://www.freedesktop.org/software/appstream/docs/chap-Metadata.html>

<sup>18</sup><https://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version>

<sup>19</sup><https://jwd.pages.apertis.org/apertis-website/concepts/applications/>

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

169 The `mimetypes` tag MUST NOT be present. In Apertis, content-type support is  
170 handled by [Entry points](#).

171 The `project_group` tag MUST NOT be present.

172 Tags that are not specified in the AppStream upstream XML specification  
173 MUST NOT be present, with the exception of `custom` (see [Extended bundle](#)  
174 [metadata](#), below).

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

### 177 **Extended bundle metadata**

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

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

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

187 Other keys SHOULD NOT be present.

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

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

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

## 203 AppArmor profile

204 Apertis uses [AppArmor](#)<sup>20</sup> to provide [security between application bundles](#)<sup>21</sup>. Each app-bundle MUST install exactly one AppArmor profile file at  
205 `${prefix}/etc/apparmor.d/Applications.${bundle_id}`, replacing `${bundle_id}`  
206 with the **Bundle ID**.  
207

208 This file MUST define exactly one AppArmor profile. Its name MUST be exactly  
209 `/Applications/${bundle_id}/**`, again replacing `${bundle_id}` with the bundle ID.  
210 It MUST NOT have any [local profiles](#)<sup>22</sup> (also known as child profiles or subprofiles), and in particular MUST NOT have any [hats](#)<sup>23</sup> (which are a special case  
211 of local profiles).  
212

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

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

---

<sup>20</sup><https://gitlab.com/apparmor/apparmor/wikis/About>

<sup>21</sup><https://jwd.pages.apertis.org/apertis-website/concepts/security/#security-between-applications>

<sup>22</sup>[https://gitlab.com/apparmor/apparmor/wikis/AppArmor\\_Core\\_Policy\\_Reference#local-profiles-and-hats](https://gitlab.com/apparmor/apparmor/wikis/AppArmor_Core_Policy_Reference#local-profiles-and-hats)

<sup>23</sup>[https://gitlab.com/apparmor/apparmor/wikis/AppArmor\\_Core\\_Policy\\_Reference#local-profiles-and-hats](https://gitlab.com/apparmor/apparmor/wikis/AppArmor_Core_Policy_Reference#local-profiles-and-hats)



```

240
241     signal receive peer=/usr/bin/canterbury,
242 }

```

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

245 Future direction: [the profile should be generated from simpler meta-](#)  
 246 [data](#)<sup>24</sup> in a future minor version of this specification.

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

## 252 Entry points

253 Each app-bundle MAY install `.desktop` files in the `$(prefix)/share/applications/`  
 254 directory. The contents of that file are interpreted according to the [Desktop](#)  
 255 [Entry Specification](#)<sup>25</sup>.

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

260 The name of each desktop entry file, excluding the `.desktop` extension, is called  
 261 the [Entry point ID](#).

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

Field	<a href="#">Main entry point</a>	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
NoDisplay	optional	optional	true
OnlyShowIn	<a href="#">Apertis</a> ;	<a href="#">Apertis</a> ;	<a href="#">Apertis</a> ;
Path	optional	optional	optional

<sup>24</sup><https://phabricator.apertis.org/T311>

<sup>25</sup><http://standards.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html>

Field	Main entry point	Other graphical programs	Agents
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

## 266 General fields for all entry points

267 Type **MUST** be set to `Application`.

268 `OnlyShowIn` **MUST** be set to `Apertis;.`

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

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

272       The Canterbury application manager does not support those place-  
273       holders.

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

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

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

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

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

291 Translated versions of these names, such as `Name[fr]`, **MAY** be present using the  
292 `localestring`<sup>26</sup> mechanism defined in the Desktop Entry Specification.

<sup>26</sup><https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest>.

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

297 Interfaces MAY be set, for [interface discovery](#)<sup>27</sup>.

298 DbusActivatable MAY be present and set to true, as described in [D-Bus activa-](#)  
299 [tion](#).

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

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

302 The following keys MUST NOT be present:

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

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

- 314 • Actions
- 315 • Comment
- 316 • Environment
- 317 • Keywords
- 318 • TryExec
- 319 • X-Apertis-AudioChannelName
- 320 • X-Apertis-AudioResourceOwner
- 321 • X-Apertis-AudioRole
- 322 • X-Apertis-BackgroundState
- 323 • X-Apertis-BandwidthPriority
- 324 • X-Apertis-DataExchangeRules (obsolete)
- 325 • X-Apertis-ManifestUrl (obsolete)
- 326 • X-Apertis-SettingsIcon (set an Icon instead)
- 327 • X-Apertis-SettingsName (set the name in the [Bundle metadata](#) instead)
- 328 • X-Apertis-SettingsPath (use the mechanism described in [GSettings](#)  
329 [schemas](#) instead)
- 330 • X-Apertis-SplashScreen
- 331 • X-Apertis-WindowName (obsolete)

---

html#localized-keys

<sup>27</sup>[https://jwd.pages.apertis.org/apertis-website/concepts/interface\\_discovery/](https://jwd.pages.apertis.org/apertis-website/concepts/interface_discovery/)

332 Future directions:  
333 X-Apertis-AudioRole, X-Apertis-BackgroundState and X-Apertis-  
334 BandwidthPriority are under consideration for a future minor version  
335 of this specification, but are not currently considered to be stable.

### 336 Entry point ID

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

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

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

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

### 348 Main entry point

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

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

### 355 Content type and URI scheme handlers

356 The **Main entry point** MAY be registered as the **content type handler**<sup>28</sup> for  
357 media types such as `audio/mpeg`, by setting `MimeType` to a list of **content types**<sup>29</sup>,  
358 each followed by a semicolon `;`. Non-main entry points MUST NOT be content  
359 type handlers. For example, a media player with support for the MP3 and  
360 RealAudio formats might use `MimeType=audio/mpeg;audio/vnd.rn-realaudio;` in  
361 its entry point.

362 The **Main entry point** MAY be registered as the handler for **URI schemes**<sup>30</sup> such  
363 as `tel` or `http`, by including an entry in `MimeType` for the pseudo-content-type `x-`  
364 `scheme-handler/scheme`, for example `x-scheme-handler/http` for a web browser.  
365 Non-main entry points MUST NOT be URI scheme handlers.

<sup>28</sup>[https://jwd.pages.apertis.org/apertis-website/concepts/content\\_hand-over/](https://jwd.pages.apertis.org/apertis-website/concepts/content_hand-over/)

<sup>29</sup><https://www.iana.org/assignments/media-types/media-types.xhtml>

<sup>30</sup><https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml>

366 If the entry point implements **D-Bus activation**, sending the `org.freedesktop.Application.Open`  
367 `method call`<sup>31</sup> to the object path corresponding to its entry point ID MUST  
368 result in it attempting to open the URI or URIs passed as parameters.

## 369 Graphical programs

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

373 A graphical program MAY have more than one entry point, to appear in menus  
374 more than once (for example, [the Frampton media player uses this](#)<sup>32</sup> to appear  
375 three times under the names Artists, Albums and Songs).

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

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

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

383 `Categories` MUST be set to a list of appropriate menu categories from the  
384 freedesktop.org [Desktop Menu Specification](#)<sup>33</sup>, each followed by a semicolon  
385 `;`. There MUST be at least one Main Category.

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

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

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

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

399 `Icon` MUST be set to the name of either the **Icon for the bundle** or one of  
400 the **Icons for entry points**, as a bare icon name (without any `/` characters, and

<sup>31</sup><https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus>

<sup>32</sup><https://gitlab.apertis.org/appfw/frampton/tree/v0.6.1/scripts>

<sup>33</sup><http://standards.freedesktop.org/menu-spec/latest/apa.html>

401 without a file-type extension such as `.png` or `.svg`). In particular, this implies  
402 that its string value MUST match either the **Bundle ID**, or the **Entry point ID**  
403 of an entry point.

#### 404 **Multiple views**

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

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

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

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

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

#### 423 **D-Bus activation**

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

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

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

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

435 The service activation command line MUST be a command-line that will  
436 start the program without opening any graphical windows, such that it will  
437 be ready to receive D-Bus requests. If a program uses the **GApplication**<sup>34</sup>

---

<sup>34</sup><https://developer.gnome.org/gio/stable/GApplication.html>

438 API (which is recommended), then the service activation command line for  
439 graphical programs will typically be the absolute path of the executable  
440 followed by a space and the `--gapplication-service` argument, for example  
441 `X-Apertis-ServiceExec=/Applications/com.example.ShoppingList/bin/main --`  
442 `gapplication-service`, while the service activation command line for agents and  
443 other non-graphical programs (with the `G_APPLICATION_IS_SERVICE` flag<sup>35</sup>) will  
444 typically just be the path to the executable.

445 When the service activation command line for a graphical program is launched,  
446 the resulting process MUST export a D-Bus object path that is derived from  
447 the entry point ID by prepending `/` and replacing each `.` with `/`, then request  
448 a well-known name equal to the entry point ID. The interfaces of that ob-  
449 ject path MUST include at least the `org.freedesktop.Application` interface<sup>36</sup>,  
450 and MAY include additional standard or non-standard interfaces such as the  
451 `org.gtk.Application` interface<sup>37</sup> used by Glib's `GApplication`<sup>38</sup> objects. When  
452 launched in this way, the process MUST NOT behave as though any of its en-  
453 try points were activated until it receives an appropriate D-Bus method call; in  
454 particular, it MUST NOT open any windows until it is told to do so.

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

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

465 For graphical programs, sending the `org.freedesktop.Application.Activate`  
466 `D-Bus method call`<sup>39</sup> to one of the object paths described above MUST  
467 result in the program displaying whatever window is appropriate for the  
468 corresponding entry point. If the graphical program implements [content-type  
469 handling][Content type and uri scheme handlers], then the same is true for the  
470 `org.freedesktop.Application.Open` method<sup>40</sup>. This requirement is not applicable  
471 to agents and other non-graphical programs.

<sup>35</sup><https://developer.gnome.org/gio/stable/GApplication.html#G-APPLICATION-IS-SERVICE:CAPS>

<sup>36</sup><https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus>

<sup>37</sup><https://wiki.gnome.org/Projects/GLib/GApplication/DBusAPI>

<sup>38</sup><https://developer.gnome.org/gio/stable/GApplication.html>

<sup>39</sup><https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus>

<sup>40</sup><https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus>

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

474 When the `Exec` command of a D-Bus-activatable graphical entry point is  
475 launched, the resulting process MUST arrange for a program to be run (directly  
476 or indirectly) that will request the well-known name corresponding to that  
477 entry point ID, export the corresponding D-Bus object path, and behave as  
478 though that object path had received an `Activate` or `Open` method call, modified  
479 according to the command-line arguments if appropriate: in other words, it has  
480 behaviour similar to the `Exec` command of a non-D-Bus-activatable graphical  
481 program. If the program uses the [GApplication](#)<sup>41</sup> API, this will normally  
482 be achieved by setting `Exec` to the absolute path of the executable, with no  
483 arguments, for example `Exec=/Applications/com.example.ShoppingList/bin/main`.  
484 This requirement is not applicable to agents and other non-graphical programs.

485 See the [specification of the Application interface](#)<sup>42</sup> for more details about its  
486 methods.

## 487 Agents

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

489 Agents MUST set `x-Apertis-Type` to `agent-service`.

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

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

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

## 495 Paths for other file types

### 496 Executables

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

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

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

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

503 Suitable directories are conveniently available as `${bindir}`,  
504 `${libexecdir}` and `${pkglibexecdir}`<sup>4</sup> when using Automake.

---

<sup>41</sup><https://developer.gnome.org/gio/stable/GApplication.html>

<sup>42</sup><https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html#dbus>



505 **Libraries**

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

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

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

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

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

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

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

538 For each native ELF library, the app-bundle MUST contain a file whose name  
539 exactly matches the `SONAME`<sup>43</sup> (ELF `DT_SONAME`) of the library, in a directory  
540 that will be searched by all executables that use that library (for example via  
541 `DT_RPATH` or `LD_LIBRARY_PATH`). This file MUST either be a regular file (the library  
542 itself), or a symbolic link to the library’s “real name”.

543 Building and installing shared libraries using GNU libtool is RECOMMENDED:

<sup>43</sup><http://tldp.org/HOWTO/Program-Library-HOWTO/shared-libraries.html>

544 libraries built like this will typically have a correct symbolic link from the `SONAME`  
545 to the “real name” without further action from the developer.

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

### 550 **Icon for the bundle**

551 The app-bundle MAY have an icon to represent the bundle as a whole, in a  
552 generic user interface icon theme. The generic user interface icon theme is rep-  
553 resented by the reserved theme name `hicolor`, as required by the freedesktop.org  
554 [Icon Theme Specification](#)<sup>44</sup>.

555 If the app-bundle has this icon, it MUST be in [Portable Network Graphics](#)<sup>45</sup>  
556 format, 64×64 pixels in size, and MUST be located at

```
557 ${prefix}/share/icons/hicolor/64x64/apps/${name}.png
```

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

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

562 To minimize display artefacts caused by resizing, the app-bundle MAY make this  
563 icon available in some or all of the additional sizes used in the freedesktop.org  
564 [reference implementation](#)<sup>46</sup> of the `hicolor` fallback theme (8, 16, 22, 24, 32, 36,  
565 42, 48, 64, 72, 96, 128, 192, 256 or 512 pixels). These MUST be installed to the  
566 corresponding path with `64x64` replaced by the appropriate size.

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

```
573 ${prefix}/share/icons/net.example.Metallic/64x64/apps/${name}.png
```

### 574 **Icons for entry points**

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

---

<sup>44</sup><https://specifications.freedesktop.org/icon-theme-spec/icon-theme-spec-latest.html>

<sup>45</sup><https://tools.ietf.org/html/rfc2083>

<sup>46</sup><https://www.freedesktop.org/software/icon-theme/releases/>

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

## 580 **Icons for use by the bundle**

581 The app-bundle MAY contain other icons. They SHOULD be arranged accord-  
582 ing to the freedesktop.org **Icon Theme Specification**<sup>47</sup>.

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

```
587 ${prefix}/share/icons/hicolor/24x24/actions/mail-mark-important.png
```

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

```
590 ${prefix}/share/icons/net.example.Metallic/24x24/actions/mail-mark-important.png
```

591 The app-bundle MAY assume that it will be launched with the `XDG_DATA_DIRS`  
592 **environment variable**<sup>48</sup> set to a value that includes `${prefix}/share`, so that com-  
593 mon icon theme implementations such as `GtkIconTheme`<sup>49</sup> will automatically  
594 use icons from the `${prefix}`.

## 595 **Theme data for use by the bundle**

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

598 The app-bundle MAY assume that it will be launched with the `XDG_DATA_DIRS`  
599 **environment variable**<sup>50</sup> set to a value that includes `${prefix}/share`, so that  
600 common theme implementations such as `GtkCssProvider`<sup>51</sup> will automatically  
601 use theme data from the `${prefix}`.

## 602 **GSettings schemas**

603 **GSettings schemas**<sup>52</sup> are used for **preferences**<sup>53</sup>.

604 The app-bundle MAY install one or more **GSettings schemas**<sup>54</sup> into  
605 `${prefix}/share/glib-2.0/schemas/`. The filenames used MUST be the schema  
606 ID followed by `.gschema.xml`, optionally accompanied by enum definitions in a

<sup>47</sup><https://specifications.freedesktop.org/icon-theme-spec/icon-theme-spec-latest.html>

<sup>48</sup><https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html>

<sup>49</sup><https://developer.gnome.org/gtk3/stable/GtkIconTheme.html>

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

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

<sup>52</sup><https://developer.gnome.org/gio/stable/GSettings.html>

<sup>53</sup>[https://jwd.pages.apertis.org/apertis-website/concepts/preferences-and-persistence/  
#preferences-approach](https://jwd.pages.apertis.org/apertis-website/concepts/preferences-and-persistence/#preferences-approach)

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

607 file named by the schema ID followed by `.enums.xml`. Each schema ID SHOULD  
608 either match the **Bundle ID** exactly, or start with the bundle ID followed by a  
609 dot.

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

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

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

622 The app-bundle MAY assume that it will be launched with the `XDG_DATA_DIRS`  
623 **environment variable**<sup>55</sup> set to a value that includes `${prefix}/share`, so that  
624 GSettings will automatically use these schemas.

## 625 Localized strings

626 Some file formats, such as `.desktop` files and AppStream XML, put **localized**  
627 **strings**<sup>56</sup> in a single file, typically built from an international English version  
628 and a set of translations at build-time. For the following file formats, the app-  
629 bundle MUST include all of its supported translations (for example a translated  
630 Name) in a single file:

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

633 Otherwise, application bundles that contain localized strings SHOULD use GNU  
634 **gettext**<sup>57</sup> `.mo` files. These SHOULD be stored in the `${prefix}/share/locale`  
635 hierarchy, with a subdirectory named for the *locale* in which the language is  
636 used, and a `LC_MESSAGES` subdirectory inside that containing one or more `.mo`  
637 files. The name of the `.mo` files (the *text domain*) SHOULD either be exactly  
638 the **Bundle ID**, or the bundle ID followed by a dot and one or more additional  
639 components. Using a single text domain whose name is exactly the bundle ID  
640 is RECOMMENDED.

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

---

<sup>55</sup><https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html>

<sup>56</sup><https://jwd.pages.apertis.org/apertis-website/architecture/internationalization/>

<sup>57</sup><https://www.gnu.org/software/gettext/manual/index.html>

```
644     • /Applications/com.example.ShoppingList/share/locale/fr/LC_MESSAGES/com.example.ShoppingList.mo
645     • /Applications/com.example.ShoppingList/share/locale/fr_CA/LC_MESSAGES/com.example.ShoppingList.mo
646     • /Applications/com.example.ShoppingList/share/locale/uz@cyrillic/LC_MESSAGES/com.example.ShoppingList.mo
```

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

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

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

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

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

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

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

## 672 Generic resource data

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

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

680 The app-bundle MAY assume that it will be launched with the `XDG_DATA_DIRS`  
681 **environment variable**<sup>58</sup> set to a value that includes ``${prefix}/share`, so that

<sup>58</sup><https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html>

682 any library that uses that variable (for example via `g_get_system_data_dirs()`)  
683 will automatically load resource files from the appropriate subdirectory of  
684 `${prefix}/share`.

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

## 687 Example

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

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

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

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

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

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

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

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

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

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

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

745 [Entry point][Entry points] for the agent, to be installed as `${prefix}/share/applications/net.example.ShoppingLi`

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

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

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

```

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

```

## 779 Future directions

780 Future versions of this specification could include layout and contents spec-  
781 ifications for particular categories of [system extensions](#)<sup>59</sup>, in particular user-  
782 installable UI themes and language packs.

## 783 Appendix: built-in application bundles

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

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

## 792 Structural differences

793 In general, [built-in application bundles](#)<sup>60</sup> MUST have a structure analogous to  
794 store application bundles, replacing `/Applications` with `/usr/Applications` in all  
795 path prefixes. In particular, the `/${prefix}` for a built-in application bundle is  
796 `/usr/Applications/` followed by the bundle ID.

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

<sup>59</sup><https://jwd.pages.apertis.org/apertis-website/concepts/applications/#system-extensions>

<sup>60</sup><https://jwd.pages.apertis.org/apertis-website/concepts/applications/#software-categories>



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

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

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

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

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

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

## 827 **Permissions and policy differences**

828 Recommendations and requirements that refer to the app-store curator do not  
829 apply to built-in application bundles. The platform vendor has total control  
830 over both the [platform layer](#)<sup>61</sup> and the built-in application bundles that are  
831 packaged with it; they are responsible for ensuring that those components fit  
832 together correctly and meet their functional and security requirements. For  
833 example, a platform vendor can provide any **AppArmor profile** for a built-in  
834 application bundle, and it is up to the platform vendor to ensure that the profile  
835 is consistent with their security policy.

<sup>61</sup><https://jwd.pages.apertis.org/apertis-website/concepts/applications/#software-categories>

836 **Graphical programs**

837 Built-in application bundles do not necessarily need to provide their own user  
838 interfaces if they rely on an underlying service, for example one that is running  
839 in the [automotive domain](#)<sup>62</sup>, to display a user interface. Where this specification  
840 calls for a particular entry point to be a graphical program, that requirement  
841 or recommendation does not apply to built-in application bundles. A built-in  
842 application bundle could provide similar functionality by communicating with  
843 other processes, either locally or in the automotive domain, and arranging for  
844 those other processes to display graphics instead.

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

847 **Command line arguments**

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

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

---

<sup>62</sup><https://jwd.pages.apertis.org/apertis-website/concepts/inter-domain-communication/#automotive-domain>