Sharing

Contents

Use cases

App bundles'functionality may include a feature of the form "send file to another user". Suppose the user is viewing a file in some application, either built-in or a store application.

Sharing menu

- The current application must be able to obtain a list of potential sharing recipients from the platform, sufficient to display a sharing menu similar to the one in Android. This list could include actions such as "send via MMS", "attach to an email", "send via Skype"or "share on Facebook", each with the icon of its associated app bundle.
  - *One possible way to obtain this list would be via Interface discovery*[1].
- When the user selects an item from that menu, the current application must send the location of the current file to the platform, which will respond by launching the app-bundle corresponding to the menu item.
- The platform must make the relevant file available to the receiving application as described in Content hand-over/Use-cases#Ensuring that the relevant file is accessible[2].
  - *In practice the implementation is likely to be the same as in that use-case.*
- An app bundle that offers similar functionality must be able to register to appear in this menu for all files, for instance by including a suitable entry in its manifest. This would be appropriate for a third-party email client, for example.
- Optionally, an app bundle might be able to register to appear in this menu for a subset of files selected by media type, such as "JPEG images"or "all images". This would be appropriate for a Flickr uploader, for example.
  - *Is this required?*
- App bundles whose action cannot be described as sending a file to another person, such as a PDF viewer's ability to open PDF files, should not normally participate in the sharing menu. Merely opening a file for viewing or editing should be treated as Content hand-over use-cases[3] instead.

[1] https://apertis-website-0b3586.pages.apertis.org/concepts/archive/application_framework/interface_discovery/

[2] https://apertis-website-0b3586.pages.apertis.org/concepts/archive/application_framework/content_hand-over_use-cases/#ensuring-that-the-relevant-file-is-accessible

[3] https://apertis-website-0b3586.pages.apertis.org/concepts/archive/application_framework/content_hand-over_use-cases/#opening-a-file-by-type

- A recommended user-interface toolkit library such as Mildenhall could provide a ready-made implementation of the sharing menu.

Sharing button

Another possible UX would be for the current application to have a single "Share" button, which opens a choice of possible recipients that is outside the application' s control.

- If supported, when the user activates that button, the current application must send the location of the current file to the platform, which will respond by listing all relevant sharing recipients (the same as in sharing menu) and presenting a platform-controlled UI component such as a pop-up window, which behaves similarly to sharing menu.
- Registration requirements would be the same as for sharing menu.
- "Is this a requirement? Do we need to support this?'
  - "Note that if the application cannot determine whether there are *any* possible recipients, this UX does not give it a way to disable its Share button, likely leading to user frustration when they press an apparently active button that cannot actually do anything useful. As a result, we do not recommend implementing this."