Share for iOS and Android

Share provides functionality for sharing text, images, videos, and other files through the native sharing interface on both iOS and Android platforms.

Core Functionality

Methods:

- `static void ItemsAsync(List<string> items)`

  Asynchronous version of Items().

- `static void Items(List<string> items, Action<bool> callback)`

  Shares the specified items (text, URLs, or file paths) through the native sharing interface. A native share window will open.

Share Items

Accepted Item Types:

- **Text** (simple string): Shared as plain text message.
- **URL** (string starting with `http://` or `https://`): Shared as a clickable link.
- **File Path** (e.g., `/storage/emulated/0/Download/file.jpg`): Shared as an attachment if the file exists.

Note:
You can mix different types of content (text, files, URLs) in a single sharing operation. However, support for mixed content depends on the target application. Some apps may only accept specific types of shared content (e.g., only text or only files).

Result Callback

The callback provided to `Items` indicates whether the share operation was completed or failed.

Callback Parameter:

- `bool success`
  - `true` - if the share window was successfully opened and the user returned to the app.
  - `false` - if an error occurred while trying to open the share window.

Note:

The plugin cannot detect whether the user actually completed the share (i.e., sent, posted, or canceled inside the external app).

It only detects that the share window was opened.

Platform Differences

iOS:

- Uses `UIActivityViewController`.
- Supports text, URLs, images, videos, PDFs, and other common file types.

Android:

- Uses `Intent.ACTION_SEND` or `Intent.ACTION_SEND_MULTIPLE`.

- Once the user picks an app, no further feedback is possible about the user's action.

Usage Example

Using callback:

```
Share.Items(new List<string>
{
   "Hello from Unity!",
   "https://example.com",
   Path.Combine(Application.persistentDataPath, "image.png")
}, success ⇒
{
   if (success)
   {
      Debug.Log("Share operation completed (window was opened and returned)");
   }
   else
   {
      Debug.LogWarning("Failed to open share window");
   }
});
```

Attachment Guidelines

iOS:

- Supports attachments from internal storage (`Application.persistentDataPath`), bundle resources, and external accessible locations.
- Recommended to use `Application.persistentDataPath` for cross-platform compatibility.

Android:

- Attachments must reside in the app's private internal storage (`Application.persistentDataPath`).
- Public external files (e.g., `/Download`, `/Pictures`) are not directly accessible unless copied to internal storage.