Sample Code

Supporting Drag and Drop Through File Promises

Receive and provide file promises to support dragged app files and pasteboard operations.

Download

macOS 10.15+ | Xcode 13.3+

Overview

This sample code project creates a meme generator that combines an image and text into a composite meme image which can be exported as a JPEG. The sample shows you how to support both file URLs and file promises when accepting images from Mail, Safari, Photos, and other apps that support drag and drop. It also demonstrates how to provide file promises to other apps.

Your app may need to support file promises because drag and drop on macOS uses an NSPasteboard, which is separate from the normal copy-and-paste clipboard. When dragging content that isn’t yet a file on disk at the time dragging began, an app may put a file promise onto the pasteboard instead of a file URL, to avoid blocking the main thread. File promises are different from URLs because they can be read and written asynchronously at a later time, on a background queue separate from the main queue. macOS Sierra introduced modern APIs to handle file promises.

Configure the Sample Code Project

To see this sample in action, build and run the project, then drag an image from another app or location, such as Finder, Mail, or Safari, into the app window. Dragging the image into the window imports it into a form the sample app can parse and consume. Add text to the image by clicking on the text button in the upper-right corner and typing. Reposition the text box by dragging and dropping it within the app window. Dragging the composite image outside the app window exports it as a JPEG.

Below are the various ways a drag and drop can occur that will involve either a `NSFilePromiseProvider` or a `NSFilePromiseReceiver`.

Drag sent by `NSFilePromiseProvider`:

- Drag an image out to the Finder.
- Drag an image out to Mail.
- Drag an image out to Photos.

Drag received by `NSFilePromiseReceiver`:

- Drag an image in from Safari (from a website with a jpg, png, etc).
- Drag an image in from Photos.
- Drag in a very large photo from Photos app — helps test the progress UI when receiving large photos.

In addition, a drag and drop involves just the image file's URL, without a file promise:

- Drag an image from Finder.
- Drag an image from a Mail attachment.

Support Image Import by Accepting File Promises

When setting up the view that supports drag and drop, register drag types provided by `NSFilePromiseReceiver`. This registration allows the view to accept file promises that handle dragged images from Safari or Mail. The sample does this in the first view controller's `viewDidLoad`.

```
view.registerForDraggedTypes(NSFilePromiseReceiver.readableDraggedTypes.map { NSPast
```

Handle file promises before handling URLs, because the file promise generally represents the higher-quality image and should take precedence when both types are supported. This sample provides a background operation queue, so the read/write operation doesn't block the main thread. Until the file promise is fulfilled, this sample shows a spinner or loading indicator in the image view subclass to give the user immediate feedback that the app is processing the drop.

```
case let filePromiseReceiver as NSFilePromiseReceiver:
    self.prepareForUpdate()
    filePromiseReceiver.receivePromisedFiles(atDestination: self.destinationURL, opt
                                                    operationQueue: self.workQueue) { (file
        if let error = error {
            self.handleError(error)
        } else {
            self.handleFile(at: fileURL)
        }
```

This sample continues to handle file URLs, in case the app from which a user drags the image doesn't provide file promises.

```
case let fileURL as URL:
    self.handleFile(at: fileURL)
```

Support Image Export by Providing File Promises

This sample creates and writes images into formats that other apps like Safari, Mail, and Finder can consume. It does this by providing an `NSFilePromiseProvider` instance to the dragging pasteboard and conforming to `NSFilePromiseProviderDelegate` by implementing three delegate methods.

This sample uses the first method to provide the title of the file. It uses a hard-coded string for simplicity, but depending on use case, take the `fileType` parameter into account.

```
func filePromiseProvider(_ filePromiseProvider: NSFilePromiseProvider, fileNameForTy
    let droppedFileName = NSLocalizedString("DropFileTitle", comment: "")
    return droppedFileName + ".jpg"
}
```

This sample provides a background operation queue in `operationQueue(for:)` so the file reading and writing happen without blocking the app's UI. This method is optional, but defaulting to the main queue can block an app's UI for writing large files to disk. When possible, provide a background queue.

```
func operationQueue(for filePromiseProvider: NSFilePromiseProvider) -> OperationQueu
    return workQueue
}
```

The third delegate method performs the actual writing of the file to disk when it's time to fulfill the file promise. Add custom logic necessary to transform the image from your app into a file format that other apps are likely to understand, such as the JPEG format this sample uses.

```
func filePromiseProvider(_ filePromiseProvider: NSFilePromiseProvider, writePromiseT
    do {
        if let snapshot = filePromiseProvider.userInfo as? ImageCanvas.SnapshotItem
            try snapshot.jpegRepresentation?.write(to: url)
        } else {
            throw RuntimeError.unavailableSnapshot
        }
        completionHandler(nil)
    } catch let error {
        completionHandler(error)
    }
}
```

See Also

File Promises

{} Supporting Table View Drag and Drop Through File Promises

Share data between macOS apps during drag and drop by using an item provider.

{} Supporting Collection View Drag and Drop Through File Promises

Share data between macOS apps during drag and drop by using an item provider.

`class NSFilePromiseProvider`

An object that provides a promise for the pasteboard.

`protocol NSFilePromiseProviderDelegate`

A set of methods that provides the name of the promised file and writes the file to the destination directory when the file promise is fulfilled.

`class NSFilePromiseReceiver`

An object that receives a file promise from the pasteboard.