

Diffusion

Social networking made easy

Features

Share text, links, and files on iOS!

Supported platforms:

Uses the native iOS sharing menu to give access to all of the built in social networks. Compatible with iOS 6 and higher.

- Facebook
- Twitter
- Email
- Messages
- Camera Roll (for photos)
- ..and more!

Easy integration

- Configure your messages and what platforms to share on through the inspector
- Start the sharing process with a single line of code!
- Check whether users are already logged into Facebook or Twitter
- Includes callbacks for when sharing is successful, or cancelled

Extensions

Diffusion supports iOS 8 Extensions and automatically includes any additional platforms the user has installed on their device.

There is also a simple interface for adding your own custom UIActivities for whatever platforms you may want to support!

Upgrade Notes

Caution

Diffusion 3.0 requires Unity 5, and uses the build in Framework import tool. Double check that Diffusion.m and Diffusion.h have the Social framework enabled.

Support

For support or to report any issues, please contact us at support@scarlet.io or [@scarlet_io](https://twitter.com/scarlet_io) on Twitter.

Getting Started

Install the Unity package and drop the Diffusion prefab into your scene, then you're ready to go.

Send a `Share()` message to the prefab to share your content!

Configuration

Platforms to hide

The networking menu is generated automatically based on what type of data you want to share, and what platforms are available. If the user has not signed into Facebook, it will not show up in the list, for example. However, sometimes options will show up that you do not actually want. For example if you're only sharing text, you may get a "Copy" option which isn't very useful.

This selection will allow you to explicitly hide options that you do not want to show. Many are disabled by default.

Custom platforms

Diffusion offers an interface for using external UIActivities. To use them, you need to include the files provided by the UIActivity into `Plugins/iOS` in your Unity project, and add the class to the Custom Platforms field on the Diffusion prefab.

In order for Diffusion to access the classes for your custom UIActivity, you'll need to create a new category that includes the headers. You can do this through the new file menu in Xcode, or by hand.

Methods

`Share()`

Shares the message and URL set in the inspector.

`Share(string message, string filePath)`

Shares the message provided, and a file.

`Share(string message, string url, string filePath)`

Shares a message, URL, and file as provided.

`isFacebookConnected()`

Checks if the user has logged into Facebook.

`PostToFacebook()`

Posts the message and URL set in the inspector to Facebook.

`PostToFacebook(string message, string imagePath)`

Posts the message and image to Facebook.

`PostToFacebook(string message, string url, string imagePath)`

Posts a message, URL, and image to Facebook.

`isTwitterConnected()`

Checks if the user has logged into Twitter.

`PostToTwitter()`

Posts the message and URL set in the inspector to Twitter.

`PostToTwitter(string message, string imagePath)`

Posts the message and image to Twitter.

`PostToTwitter(string message, string url, string imagePath)`

Posts a message, URL, and image to Twitter.

Events

If you connect a GameObject into the Event Receiver field on the Diffusion prefab, it will these messages will be sent to it.

Do **not** put your event receiver script on the prefab.

`OnCompleted(DiffusionPlatform platform)`

Sharing completed successfully. Returns the platform the user selected to share with.

`OnCancelled()`

User cancelled the sharing process.

Delegates

For slightly better performance, or if multiple listeners are necessary, the above events are also exposed as delegates.

Notes

- Twitter's URL shortener assumed any URL to be 22 characters (20, plus spaces). In reality, there is a bug in the UIActivity's character counting, and URLs show as 36 characters.
 - Facebook will not prefill a message supplied through code, when selected from the share menu. This is to comply with Facebook's [Platform Policy \(2.3\)](#).
 - The `PostToTwitter()` and `PostToFacebook` methods return the same callbacks as if the user selected those options from the general share menu.
-