



Promoting a mobile application is not easy. It's getting more difficult every day as the market gets flooded with new apps, the majority of which are crap. It's very, very easy to get lost in the market noise.

The basic pattern that most of us follow in releasing a commercial application looks something like this:

1. Create a "Lite" version of an application. Release to market, promote, get reviewed, tweak, improve, develop strong and loyal user base.
2. At some point in the future, release a "Professional" version of the application and charge for it. Add compelling features not available in Lite version, and make it easy to upgrade for existing user base.

I've had some interesting insights on doing this with my application [Screabl](#), and I'll be blogging on exactly what I've learned in the process in an upcoming post. I think that what I've discovered will be very useful for other devs.

One of the outcomes of writing [Screabl](#) has been my finally "getting" the whole [Twitter](#) thing. I've toyed with Twitter for years now, but supporting an app like [Screabl](#), getting feedback, and disseminating/promoting frequent updates on progress is absolutely ideal for Twitter. It turns out that it is also very difficult to develop a Twitter subscription base.

I recently made the acquaintance of a very bright developer that is part of the team responsible for the Android application called [Swift Twitter App](#). We both attended the Google Android Developer Lab in NYC back in November. We had great fun speculating on our application's

chances in the then-yet-undecided Android Developer Challenge 2. Neither of us ended up winning.

Since then, I've kept in loose contact with Sebastian, and have watched his app continue to grow and evolve. Swift is currently ranked at #10 in the free social category. They've had well over 75,000 downloads, and the Swift Twitter user @swiftapp has 22,500 followers. A very enviable position for any application.

One reason that @swiftapp has so many followers is obviously that the application caters to a client group that understands and uses Twitter. But I'm still envious. I want 22,500 followers on Twitter. Screebl has been downloaded over 25,000 times, but the Twitter account that I use for Screebl, @keyesdav, has only 63 subscribers. Swift has a very powerful channel for the dissemination of information about their application, and I want that for my apps too!

In short, I want to socialize my mobile applications.

It occurred to me a few days ago that Swift and all of the other Twitter clients could help here. If all of these Twitter clients were to agree on a standard set of Intents that could be started from other apps, then any application could leverage the power of Twitter without needing to become Twitter API experts. Imagine the following use cases:

- App developers offer a "Follow Me" button in their app. When that button or menu item is selected, one of the installed Twitter clients is invoked, with all of the required information filled in to let the user follow my application's Twitter account. Whammo. Assuming that half of my users would bite, I now have tens-of-thousands of followers on Twitter.
- App developers offer a "Tweet This" feature in their apps. As new versions of my app are released, getting word out is difficult. I would love to embed capabilities in my app that would allow the user to *tweet for me*, pre-canned messages about new app features.
- Apps show a Twitter feed related to their app. I would love to show users the latest Tweets about my app, from within my app. This would allow them to see how others are using the application, etc.

I'm convinced that this is an insanely powerful form of application promotion and socialization. If you squint, you might even be able to imagine a way to make money with this. It would also be good for the Twitter client apps, as it would give them more face time with users.

Sebastian and I started scratching out a spec for an Intent-based API, and you can see the very early thoughts around that [by looking here](#) . Please let me know if you have feedback, or would be interested in pursuing such a concept from either a consumer perspective (non-Twitter app) or from a provider perspective (Twitter client).

Finally, it seems like this same concept could apply to other social channels as well (e.g., Facebook, MySpace, etc.), and perhaps there's a need for a more general API that covers all of these. I'll be interested to hear thoughts on that too.

UPDATE: I just stumbled across a venue for doing exactly what I'm picturing. www.openintents.org appears to have generalized the idea of Intent standardization. There is actually a Twidroid intent listed in their registry for sending a Tweet! I'm going to look into spinning up some standardized Intents for Twitter/socialization of applications, using their forum...