



KeyesLabs.com is proud to announce our latest innovation for the Android platform: Automatic Application Licensing (AAL). We're excited to offer this technology to Android developers, as we think it will provide a crucial tool for those that are struggling with piracy of their mobile applications. In a nutshell, AAL licenses an application by validating that it was purchased from the Android Market. It sounds simple, it is simple. We're not sure why no one has done this before.

We've been burned by piracy ourselves at KeyesLabs. [Our successful utility application called Screebl](#) (which keeps the screen on based on how you hold your phone) has been downloaded nearly 100,000 times. As the app has gained in popularity, however, we've noticed a frustrating increase in the number of installations that are pirated. At this point we are beyond 70% of all installs being stolen, and some days we top 90%. Holy crap! Yes, yes, we've heard all of the arguments about how piracy can't be fought, it's not really going to hurt you anyway, it's a free form of advertising. Whatever. We want a way to limit the theft of our intellectual property. Google hasn't given any indication that they're going to tackle the problem any time soon, so we took a look to see if there was anything that we could do.

Our requirements were pretty demanding. We evaluated the other licensing options available for Android and none of them felt right. Here's the top-ten requirements that we had in creating AAL:

1. Convert pirates into patrons. Piracy protection should be primarily focused on convincing thieves that what they've stolen is worth paying for.
2. No custom back-end server required. We've got enough going on at this point. The last

thing that we need is another system to manage and software package to install and learn.

3. Work seamlessly with the Android Market. There are some really good licensing technologies available on other markets right now, but the Android Market is where the action is at the moment.

4. No management of user license keys. I hate getting a key in email and pasting it into a \$.99 app, and my tolerance for pain is high. Users should be able to buy the app from Android Market, install it, and go.

5. Minimum footprint. Whatever solution that we came up with needed to be small and fast. We didn't want to add more than 50 KB to the footprint of our apps. The library came in at 35 KB.

6. Configurable license validation behavior. Developers need to be able to decide what happens if purchase validation fails. Lock the user out? Nag the user? How many times can validation fail before things get uncomfortable?

7. Easy installation. We want to be able to install AAL in an app by adding one jar file to the build path, and adding literally one line of code.

8. Affordable. Flat-fees. No royalties and no subscriptions. Well, for us it's free since we wrote it, but we want to keep this simple and accessible for the rest of the market too.

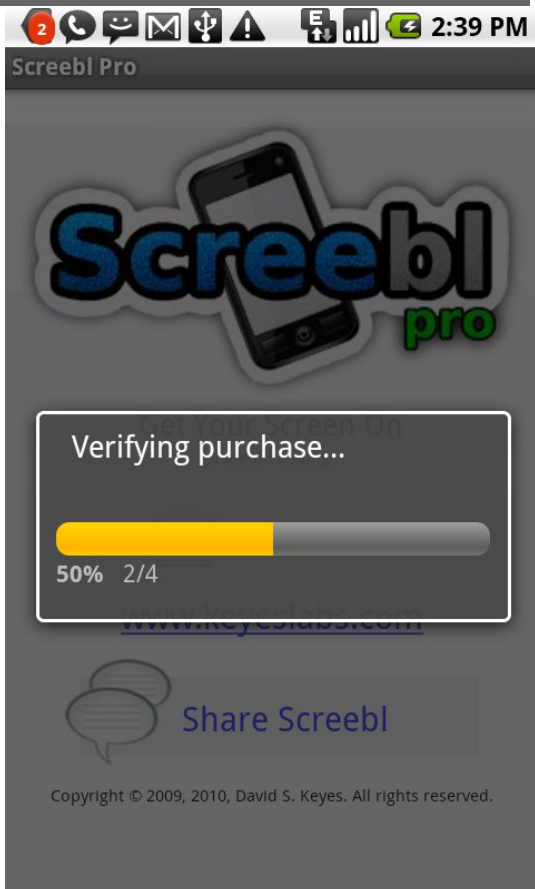
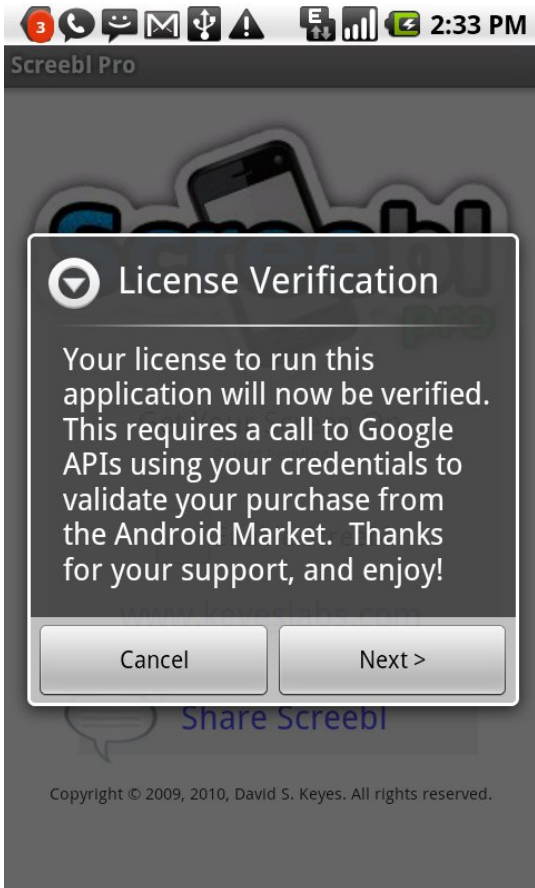
9. No requirement for users to type in user names and passwords. We've never been willing to do that for an unknown app, and we wanted to avoid it for our own apps. We're using the fancy new Account capabilities available in Android 2.0 and later.

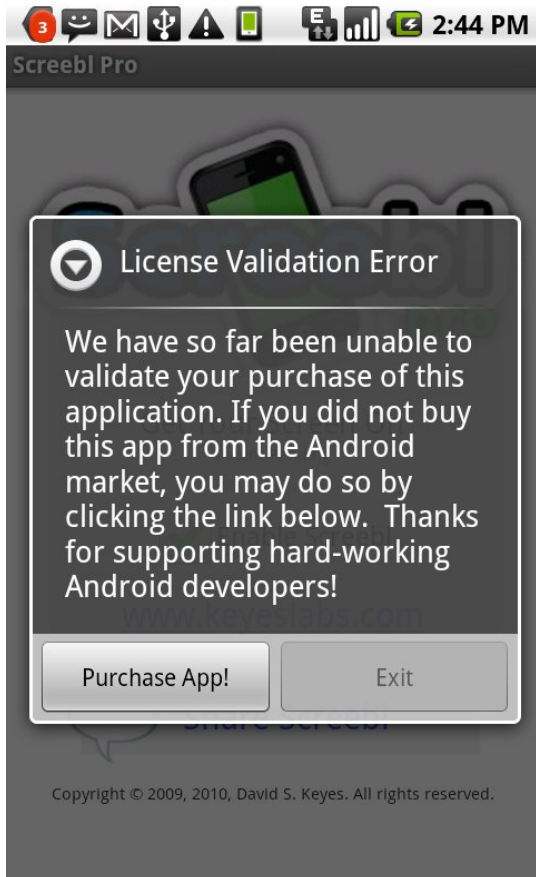
10. Compatible with all versions of Android 2.0 or later. Incompatibilities should degrade gracefully for 1.5 and 1.6.

We think we've come up with a solution that meets all of these requirements. Automatic Application Licensing is simple to use. Just add one line of code to your Activity:

```
LicenseManager.checkLicense("A6AC5F626721A935FED252F41FB14252", this);
```

The long string is a license key that enables AAL to work with your specific app. It's how we'll make money off of AAL. If you don't include it you'll get a demoware nag screen every time that you call the checkLicense method, but things will still work.





NOTE -- There have been lots of good questions in the comments. I've moved them to the forum where I can spend some time answering.