The QRdvark API

Android Background

Android is built around interoperability. An Android application like QRdvark can allow other applications to use its elements. Your app doesn't need to link to a library or include source code. You simply start up a piece of QRdvark when you need it.

If you need to scan QR barcodes, your application can call QRdvark to do the scanning and decoding and we'll pass the scanned data to your application.

Android applications are collections of components that are called by the system as needed. There are four types of Android components: *Activities*, *Services*, *Broadcast Receivers*, and *Content Providers*. It's Activities that are important when working with QRdvark.

Activities present the user with a visual interface to do something. An application often has multiple Activities, giving a user multiple things they can do (browsing your inbox, reading a message, composing a message, etc.). Using Activities gives you access to QRdvark's features. Call QRdvark from your application and you'll get a bit of QRdvark UI and functionality (encoding or decoding QR symbols).

Android Intents

Content Providers are invoked through the use of a *ContentResolver* but the other three application component types including Activities are invoked using *intents*.

An intent is a message. The contents of the message are held in an Intent object. While an intent can invoke a Service or deliver information to a Broadcast Receiver, we're interested in invoking an Activity (QRdvark) from your application.

QRdvark and Intents

QRdvark makes Activities available for scanning QR Barcodes, for encoding specific information as QR Barcodes, or launching the "share" feature of QRdvark to let the user browse and pick the information to share. Your application can invoke these activities using intents.

Scanning Symbols

You can invoke QRdvark's scanning activity with this code. It displays QRdvark's scanning screen allowing the user to scan a QR symbol. It then returns control to your via "onActivityResult".

```
public Button.OnClickListener mScan = new Button.OnClickListener() {
    public void onClick(View v) {
        Intent intent = new Intent("com.azalea.qrdvark.android.SCAN");
        startActivityForResult(intent, 0);
    }
};

public void onActivityResult(int requestCode, int resultCode, Intent intent) {
    if (requestCode == 0) {
       if (resultCode == RESULT_OK) {
           String contents = intent.getStringExtra("SCAN_RESULT");
           // Handle successful scan
       } else if (resultCode == RESULT_CANCELED) {
           // Handle cancel
       }
    }
}
```

Encoding Data

Encoding information is similar, but no result is needed.

```
public final Button.OnClickListener mEncodeURL = new Button.OnClickListener() {
    public void onClick(View v) {
        Intent intent = new Intent("com.azalea.qrdvark.android.ENCODE");
        intent.putExtra("ENCODE_TYPE", "TEXT_TYPE");
        intent.putExtra("ENCODE_DATA", "http://www.qrdvark.com");
        startActivity(intent);
    }
};
```

Sharing Info

QRdvark's "share" features can be invoked allowing users to pick an app or contact and create a symbol for someone else to scan.

```
public final Button.OnClickListener mShareViaBarcode = new
Button.OnClickListener() {
    public void onClick(View v) {
        startActivity(new Intent("com.azalea.qrdvark.android.SHARE"));
    }
}
```

Next

It is our sincere hope that others take advantage of QRdvark as an Android widget. By no means are we done. We're barcode nerds first and foremost, and we've given QR + mobile a lot of thought. Please join us in growing QR. Suggestions, corrections, shared projects, and engaged dialog. Let's play!

PN & JW, 22jul10
QR-barcodes.com/QRdvark/
azalea@azalea.com

See it.
Scan it.
See what
happens.

QRdvark