XTIFY-ANDROID 1.5

Hello fellow buzztouchers!! sorry for the long wait, but i finally found some time to put this thing togethe!! If it loks like 1.4 it's because i copied most fom that post and changed what needed to be changed, so don't be discouraged.

THIS GUIDE IS FO IMPLEMENTATION PIOR TO 11-01-2011. XTIFY HAS UPDATED THEIR IMPLEMENTATION GUIDES.

Here is what I used:

•Buzztouch 1.5 (Button Layout)
•Windows 7 32 bit
•Eclipse Version: 3.7.0
•Build id: M20110909-1335
Xtify android Sdk V1050

1.SIGN UP FOR XTIFY NOTIFICATIONS HERE http://www.xtify.com/

2.After creating your account , click on documentation within your developers console and goto Step 4: Integrate the Xtify code within your application and click on the android link.

3.Complete Section I: REGISTER YOUR APPLICATION WITH XTIFY. 2 key points of this step are Application Key Type* and Intent Filter Action name*. For the Application Key Type* put development/testing, and for the Intent Filter Action name* type android.intent.action.MAIN and then click the submit button. You should now see a app key and a secret key in your console!

4.Here's where things get a little tricky! Go back to the implementation guide for android http://console.xtify.com/implementation-guide-android SECTION II: IMPLEMENT THE XTIFY CODE INTO YOUR APPLICATION. You can download the Acme app and the sdk package from xtify or just use the sample buzz app I provided. (download) http://db.tt/tiCBMEMA It is best to use the app I provided because it has the code already implemented into a buzztouch app as opposed to the acme app! From this point on I will be referring to the sample buzztouch app.

5.Open eclipse and start a new project with the sample 1.5 buzztouch app.

6.Once the new project has opened, open up the Androidmanifest.Xml. **IMPORTANT!** Replace the two occurrences of 'YOUR_APP_KEY_GOES_HERE' with the APP key you registered earlier. Note that one of these has a preceding forward slash that must be present!

7.Run the project as android application. Now to test sending notifications from the Xtify Console, go to the "Test Implementation"

page, choose your device and click Deliver a test notification to selected users. When using the emulator, the SDK tries to access the device location. The emulator will show a 'Force close' dialog box unless you enable cell tower-based locations by checking the 'Use wireless networks' checkbox in Settings > Security & location.

YOUR APP.

First things first! The most important thing to remember is that you are only working with 3 files! Androidmanifest.xml, Act_Activitybase.java(main activity jave file), and Strings.xml. Other than that you will import the xtify sdk and copy and paste a notification icon into your project. That's it!!

1. Open your app in eclipse. First we want copy the xtify sdk package from the sample buzztouch project into your project. The 'XtifyAndroidSDK.jar' file is placed in the 'jar/' directory of the sample project. Copy the jar file from the sample buzztouch project and paste it into your jar directory. Now the jar must be added to the classpath. In Eclipse, this can be accomplished by clicking Project Properties > Java Build path > Libraries > Add JARs. Anytime you copy and paste in eclipse you want to clean the project so that the objects you pasted will be included in your project when you run it. So goto project from the top menu and clean your project now. Project-Clean, select your project

2. Goto the sample project Res folder-Drawable-and copy the notifications.png file. Paste the notifications.png into your res-drawable folder. Xtify forgot to put this valuble info into their implementation directions! Our xml file and our jave file will point to these 2 steps so it's important to do them first to avoid many errors that pop up! Unfortunately Xtify gives you the opisite way of implementing wich in turn makes you scream and pull your hair out!!

3. Open the sample app Androidmanifest.xml file and then open your project Androidmanifest.xml file. Compare them side by side. The sample xml has the xtify code implemented. First Copy and paste the permissions from the sample file and replace your permissions line 11-19 EX(A). Next, I have commented the file with xtify starts here,and xtify ends here (line 45) to make it easier to see where the code starts, and it ends at xtify ends (line 111)-EX(B) It already has your app key placed where it should be. And it already has the sdk version code already corrected so just copy and paste after:

EX(A)

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
	<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
	<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
	<uses-permission android:name="android.permission.READ_PHONE_STATE" />
	<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
	<uses-permission android:name="android.permission.INTERNET" />
	<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
	<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
```

```
        <uses-permission android:name="android.permission.VIBRATE" />
```

EX(B)

```
<intent-filter>
                    <action android:name="android.intent.action.MAIN" />
                    <category android:name="android.intent.category.LAUNCHER" />
             </intent-filter>
             </activity>

              <!-- XTIFY STRATS HERE > -->
```

Your manifest should look exactly like the sample project manifest except for the package name above <uses-sdk android:minSdkVersion=8 /> . Be sure to save you manifest file now.

4. In the sample project open up the Act_Activitybase.java file.(Src/com.v1_.5……./ Act_Activitybase.java). Now open up your Act_Activitybase.java file and again compare the two. We are not going to copy everything this time. Click on the + sign next to import org.json.JSONObject of the sample project. Notice it has 2 packages from xtify imported!(LINE 59,60).

We need to copy and paste the imports into your Act_Activitybase.java or just type them into your file yourself. Both ways import the packages.

```
import com.xtify.android.sdk.ClientMetrics;
import com.xtify.android.sdk.PersistentLocationManager;
import android.os.AsyncTask;
```

5. copy line 101 and 147 of sample project and paste it after- public String locationListenerType = ;

```
public LocationManager locationManager;
      public int locationUpdateCount = 0;
      public String locationListenerType = "";

      // Xtify location
      private PersistentLocationManager persistentLocationManager;
```

6. Now look at the sample poject. Copy line 101)// XTIFY CODE STARTS HERE - (line 126) //XTIFY CODE ENDS HERE. and paste it after //activity life-cycle events. and end before //remember this activity.

```
// ////////////////////////////////////////////////////////////////////
      // activity life-cycle events.

      // Xtify CODE STARTS HERE

      // onCreate
```

```
	@Override
	public void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		final Context context = this;

		persistentLocationManager = new PersistentLocationManager(context);

		Thread xtifyThread = new Thread(new Runnable() {
			public void run() {
				persistentLocationManager

	.setNotificationIcon(R.drawable.notification);
				persistentLocationManager

	.setNotificationDetailsIcon(R.drawable.icon);
				boolean trackLocation = persistentLocationManager
						.isTrackingLocation();
				boolean deliverNotifications = persistentLocationManager
						.isDeliveringNotifications();
				if (trackLocation || deliverNotifications) {
					persistentLocationManager.startService();

				}
			}
		});
		xtifyThread.start(); // to avoid Android's application-not-responding
		// dialog box, do non-essential work in another

		class SendMetricsTask extends AsyncTask<Void, Void, Boolean> {
			@Override
			protected Boolean doInBackground(Void... params) {
				ClientMetrics cm = new ClientMetrics(context);
				return cm.reportAppOpen();
			}

			@Override
			protected void onPostExecute(Boolean result) {
				// handle reportAppOpen results
			}
		}
		new SendMetricsTask().execute();

		// XTIFY CODE ENDS HERE

		// remember this activity...
		thisActivity = this;//remember this activity...
thisActivity = this;
```

By this point you should have no errors, only warnings. Don’t be afraid to look at the sample files side by side with yours, it’s a good way to find mistakes.

7. Finally, open up the sample strings.xml and your strings xml. Add these string to your xml.

```
      <!-- Settings Screen -->
 <string name="enableNotificationsTitle">Enable notifications</string>
<string name="enableNotificationsSummary">Enables or disables notifications
notifications</string>
<string name="locationTrackingCategoryTitle">Location Updates</string>
<string name="locationTrackingTitle">Enable location updates</string>
<string name="locationTrackingSummary">Sends location updates to the server for more
relevant  notifications</string>
<string name="locationTrackingFrequencyTitle">Usage frequency</string>
<string name="locationTrackingFrequencySummary">Specify the number of
minutes</string>
<string name="gpsUsageCategoryTitle">GPS</string>
<string name="gpsUsageTitle">Use GPS when available</string>
<string name="gpsUsageSummary">Uses GPS  when more accurate location updates are
necessary</string>
<string name="gpsUsageFrequencyTitle">Usage frequency</string>
<string name="gpsUsageFrequencySummary">Specify the number of minutes</string>

<!-- notification details screen -->
<string name="settingsButtonLabel">Settings</string>
<string name="iLikeButtonLabel">I Like</string>
<string name="iDontLikeButtonLabel">I Dont Like</string>
<string name="moreInfoButtonLabel">More Info</string>
<string name="shareButtonLabel">Share / Save</string>
<string name="mapButtonLabel">Map</string>
<string name="shareNotificationDialogTitle">Share notification via</string>
```

8. That’s it!! Now save your project and run it. Now to test sending notifications from the Xtify Console, go to the “Test Implementation” page, choose your device and click Deliver a test notification to selected users. When using the emulator, the SDK tries to access the device location. The emulator will show a 'Force close' dialog box unless you enable cell tower-based locations by checking the 'Use wireless networks' checkbox in Settings > Security & location. if you did not uninstall the test app from the virtual device our phone, you will receive two notifications because two app have the same app key!

9. remember to change your application key to Release / Production before sending notifications.