Extracted from:

Developing Android on Android

Automate Your Device with Scripts and Tasks

This PDF file contains pages extracted from *Developing Android on Android*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2013 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



Developing Android on Android

Automate Your Device with Scripts and Tasks



Mike Riley edited by Jacquelyn Carter

Developing Android on Android

Automate Your Device with Scripts and Tasks

Mike Riley

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at http://pragprog.com.

The Android robot is reproduced or modified from work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License.

The team that produced this book includes:

Jacquelyn Carter (editor) Potomac Indexing, LLC (indexer) Molly McBeath (copyeditor) David J Kelly (typesetter) Janet Furlow (producer) Juliet Benda (rights) Ellie Callahan (support)

Copyright © 2013 The Pragmatic Programmers, LLC. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America. ISBN-13: 978-1-937785-54-3

Encoded using the finest acid-free high-entropy binary digits. Book version: P1.0—November 2013 This book is dedicated to my three favorite *M*'s: Marinette, Marielle, and Mitchell.

4.3 Train Station Alarm

Those who regularly commute by train know all too well how the lull of a rocking train can put tired passengers to sleep in no time. If you're a light sleeper, you may hear the conductor announce your train stop. If not, you may wake up with the conductor tapping you on the shoulder and asking you to exit the train at the end of the line, usually a long way from home. If the train always runs on time, you could simply set a clock alarm to go off. But what if the train runs late because of rush-hour traffic conditions? Wouldn't it be nicer to get a few more minutes of a power nap in before confidently being alerted that you will be arriving shortly at your intended destination? Thanks to a relatively simple Tasker script, taking this idea from conception to reality will take only a few minutes to implement.

Creating the Alarm

The first thing we need to do is create a task that will sound an alarm. Do so by selecting the Task tab from Tasker's main screen and then create a new task by selecting the plus symbol in the bottom toolbar. Name the new task Alarm and then add the one and only action to this task by selecting the plus symbol in the new task screen. Since we want to sound an alert with this task, select the Alert action category. This will display the dialog shown in Figure 33, *The Select Alert Action dialog*, on page 6.

Choose the Notify Sound action and for this example name the action Train Stop Ahead. Tasker will display this name in Android's notification bar area when the action is triggered. If you prefer to display text other than the name of the task we assigned, you can do so by entering it in the Notify Sound optional Text field. Next, choose a sound file you prefer to play when this action is triggered. This can be an audio file in any format that your Android device can natively play back, such as a WAV or MP3 file. Select the file to be played back by tapping the magnifying glass icon in the Sound File row and navigating to and choosing the audio file you prefer. Once configured, your configuration should look similar to the one in Figure 34, *The Notify Sound dialog*.



Now that the Alarm task has been defined, we need to wrap it around a context and create a profile that will sound the alarm when a radius within a defined geographic location is entered.

Profiling the Alarm

Select Tasker's Profile tab followed by the plus symbol on the Profile toolbar. Call this new profile Train Station and choose Location from the First Context pop-up menu. A Google-powered map will appear, allowing you to place a marker at the location you want to trigger the task to run. To set the trigger marker, touch and hold the screen over the map location you want to use.

Tasker sets the default geographic radius trigger to 30 meters and sets the global GPS location sampling rate to 30 seconds. We could increase the sampling rate value located on Tasker's Monitor preference tab, as shown in Figure 22, *Network Location Check Seconds field on the Monitor tab*, on page

?, but it would also burn up our battery charge. Instead, consider how much distance a train might cover in thirty seconds and expand the trigger radius accordingly. Change this value by selecting the Radius field (this will display a list of distances in meters, as shown in Figure 35, *GPS radius list of distances*) and setting the GPS sampling radius.

In my example, I expanded this to a 300-meter radius. It's an adequate setting for my needs, since the train reduces its speed as it approaches the station. Once set, your screen will overlay a circle highlighting the event trigger area similar to the one in Figure 36, *Map with GPS radius overlay*.



Figure 35—GPS radius list of distances



Figure 36—Map with GPS radius overlay

If you need additional time to wake up and gather your items, relocate the trigger point a quarter mile (roughly 400 meters) or further from the station. Also, expand the sampling radius to 600 or more meters depending on how fast the train is traveling. I hope that Tasker's developer allows users to enter their own radius values in a future update, since I have found that the list of radius choices can be limiting at times.

Once you have set your location marker and the desired sampling radius, touch the Location Edit label to save your settings. Tasker will then ask you to name the context that you just created. Call it Sound Alarm and assign the Alarm task we created to this context. The completed profile screen should look similar to the one shown in Figure 37, *The Train Station Tasker profile*.



Figure 37—The Train Station Tasker profile

That's all there is to it. But before placing your waking trust entirely in the alert, test the task to account for train speed, music playback volume, GPS signal, and battery consumption rate.

Enhancing the Alarm

For the alarm to work, we need to make sure the GPS radio is turned on and the music playback volume is loud enough to hear. And what the heck, let's also make the Android phone vibrate to really get our attention.

Open the Alarm task and insert a new task above the existing Notify Sound action by selecting the Notify Sound step and holding down until the Action Options dialog pops up. Select Insert Action from the list. You could try to select the GPS action from the Misc Action category, but if you're running Android OS 2.3 or newer, you won't be able to unless your device has been rooted (something I don't recommend doing if you don't need to). If you

opt not to jailbreak and root your phone, you will need to remember to manually turn on the GPS for the location trigger to work.

Let's increase the media playback volume before the Notify Sound event. Select the Notify Sound step as before and select Notification Volume from the Audio Settings Action category. The default level is 3, but we'll bump that all the way up to 7, the highest level. Assuming your sound file is normalized, that sound level should definitely get your attention.

Now let's add one more activity to get our attention after the Notify Sound step. Select the plus toolbar icon in the Task Edit screen and add the Vibrate action from the Alert Action category. The default vibrate duration is 200 milliseconds. Increase that to the full 1000 milliseconds (equal to 1 second). If that isn't a long enough duration, duplicate this step for as many seconds as you need the phone to vibrate. Test the revised profile to see how it performs, and tweak the audio levels, GPS target location, and trigger radius until you consistently and reliably set off the alarm at the time and location that works best for you. Once perfected, remember to save your work!