Android Wear* through ADB

Wearables are one of the latest trends in computing technology. Google’s Android* Wear operating system makes wearables a fertile new area for app development.

This article gives an overview of the Android Wear operating system focusing on wearable devices, application types, development, and debugging. It also explains two ways of debugging a wearable app using ADB.

**Devices**

The concept of wearable computers includes different types of devices: wearable headsets, fitness and medical devices, digital jewelry, and even wearables for pets. But nowadays the leading product category is smartwatches. The biggest high tech companies offer their own lineup of wristwatches based on Android Wear. Pebble Steel*, ASUS ZenWatch*, Motorola 360*, LG G Watch R*, Samsung Gear S* are the latest on the market. All of them have different designs, but they share some common functionality supported by Android Wear: Google Now* technology, fitness tracking, controlling music, and voice commands. Also, all smartwatches depend on mobile Android/iOS* devices communicating with them through Bluetooth*. There are special companion apps for smartphones and tablets to connect to wearables.

**What to develop?**

Although Android Wear is a relatively new project, the Android Wear Center, an analog of Google Play*, provides a wide range of applications specifically designed for wearables.

A wide variety of applications are available for smartwatches. Every day Android Wear Center publishes new releases of personalization, music, communication, health, fitness, and other apps. Despite the small size of the smartwatch screen and while not yet abundant, arcade and puzzle games are available too.
The vast majority of Wear apps are watchfaces, which customize the essential wristwatch function – showing the time.

How to develop?

On one hand, creating apps for Android Wear is similar to developing for tablets and smartphones. You can use familiar development tools like JDK, Android SDK (Android Wear supports using most of the standard Android APIs), Eclipse*, Android Studio, or other IDEs. Here, you can find the list of the Wearable Support Library classes.

On the other hand, Google provides a vision and design principles unique to wearable app development that cover the essential differences between mobile and wearable technologies. The small screen size and special interaction characters are the differences that your app will have to account for. In addition, you should consider your app structure, context awareness, UI, style, and watch faces.

How to debug?

Debugging is an inherent process of any development life cycle, and developing Android Wear apps is not an exception. This section demonstrates how to debug wearable apps. Two devices are used: LG G Watch R paired with Nexus 4*.

Android Wear supports two ways to debug your device: over USB and over Bluetooth.

Regardless of which method you use to connect wearables to your PC, you need to do following initial steps:

- **Install ADB on your PC.**
  
  Android Debug Bridge (ADB) is a command line tool that provides communication between your PC and Android devices or Android device emulators.

- **Prepare devices for connection.**
  
  You need to enable the USB debugging option not only on your wearable, but on the paired mobile device too. This process is common for all Android devices: go to Settings, tap About, and then tap the build number 7 times to activate Developer Options.
Go to Developer options and enable ADB debugging.

- Connect the wearable device through the USB cable.
- Allow Wearable debugging by tapping “ok” on the pop-up window on the paired phone or tablet.
To ensure ADB connection tap “adb devices” at the command line.

```
C:\\Users\\abelova\\adb devices
List of devices attached
412KPFX0149827 device
```

The Bluetooth case is a little more complicated:

- Enable Debug over Bluetooth on wearable:

```
  Bluetooth snoop
  logging
  Disabled
  ADB debugging
  Enabled
  Debug over Bluetooth
  Disabled
  Revoke debugging authorisations
```

- In the Android Wear companion app enable Debug over Bluetooth. You can see the status under the option:
- Connect the phone or tablet paired with the wearable to the PC through USB cable and allow USB debugging.

- Tap the following commands:
  
  `adb forward tcp:4444 localabstract:/adb-hub`
  
  `adb connect localhost:4444`

- Allow Wearable debugging:
After this, the status will change to:

When your connection is successful, a list of devices, like the one shown below, displays:

Now all steps are complete, and you can use ADB commands to debug your app.

**How to take screenshots?**

ADB is useful for other things in addition to debugging. Taking screenshots on wearables is not as trivial as it seems. The “Take wearable screenshot” option in an Android Wear companion app allows only sharing screenshots through mail or social networks. You can use ADB as another way to save images of wearable screens on your PC.

```
adb shell screencap -p /sdcard/screenshot.png
adb pull /sdcard/screenshot.png
```

Notice that even on round dials the screenshots are actually square. You should keep this point in mind to improve your apps’ usability.
Summary

The combination of the modern technologies like Intel Quark processors and the Android Wear operating system opens up new opportunities for application development. As you can see, Android developers experienced in creating apps for the mobile industry can easily shift to creating apps for wearables like smartwatches, taking care of some nuances.

References

- Android Wear
- Debugging over Bluetooth

About the Author

Anna Belova works as a Software Engineering Intern in the Software & Services Group at Intel Corporation. She is getting a bachelor degree in Business Informatics at the National Research University Higher School of Economics, Business Informatics and Applied Mathematics Faculty. Anna is interested in mobile technologies and machine-learning.

Notices

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, and Intel Quark are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

© 2015 Intel Corporation.