



Creating ANT+ Android Applications

D00001446 Rev 4.0.0

P +1 403.932.9292 F +1 403.932.4196

Connecting Sensors for Life!

Copyright Information and Usage Notice

This information disclosed herein is the exclusive property of Dynastream Innovations Inc. No part of this publication may be reproduced or transmitted in any form or by any means including electronic storage, reproduction, execution or transmission without the prior written consent of Dynastream Innovations Inc. The recipient of this document by its retention and use agrees to respect the copyright of the information contained herein.

The information contained in this document is subject to change without notice and should not be construed as a commitment by Dynastream Innovations Inc. unless such commitment is expressly given in a covering document.

The Dynastream Innovations Inc. ANT Products described by the information in this document are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Dynastream product could create a situation where personal injury or death may occur. If you use the Products for such unintended and unauthorized applications, you do so at your own risk and you shall indemnify and hold Dynastream and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Dynastream was negligent regarding the design or manufacture of the Product.

©2013 Dynastream Innovations Inc. All Rights Reserved.

Table of Contents

1	Overview.....	4
2	ANT Application Requirements.....	4
2.1	Android Device ANT functionality	4
2.1.1	ANT Radio Service	4
2.1.2	ANT USB Service	4
2.2	ANT Libraries.....	4
2.2.1	PluginLib	4
2.2.2	Android_AntLib	5
2.3	ANT+ Membership and License Agreement.....	5
3	General Overview of Programming with the ANT Libraries	5
3.1	Overview of Using the ANT+ Plugins	5
3.2	Overview of Using the Android_AntLib	5
4	Sample Code	Error! Bookmark not defined.
4.1	ANT+ Plugin Sampler application and source code	5
4.2	ANT Multichannel Proof application and source code.....	6
5	Development Tools.....	6
5.1	Android SDK, Eclipse IDE and Android ADT	6
5.2	ANT Android Emulator Bridge	7
6	Reference and Support	7
6.1	ANT+ Device Profile Documentation	7
6.2	ANT Message Protocol and Usage Document	7
6.3	ANT+ Forums	7
6.4	Android Developers Guide	7
7	Notes on Use of ANT+ Logos and Certification	8

1 Overview

The purpose of this document is to provide developers all the information (or the location of information) they will need to create applications which will run on the Android platform and make use of the ANT wireless functionality available on certain devices.

The list of manufacturers who are releasing phones with ANT support, and the number of Android devices with ANT support is constantly growing. The ANT USB Service has also been released, providing ANT support on Android devices with the USB Host mode feature through the use of attached ANT USB sticks. Additionally, Android Emulator support for ANT has arrived, allowing anyone with a PC and an ANT USB stick to develop ANT applications for Android even without a physical device.

More information on ANT, ANT+ and related downloads and documentation can be found on the ANT Wireless website, <http://www.thisisant.com>.

2 ANT Application Requirements

2.1 Android Device ANT functionality

In order to utilize ANT on an Android, ANT radio support is required on the target device. This can be present as a built-in feature of the Android Device, or via an externally attached accessory such as an ANT USB stick attached to a phone with USB-Host support. A list of officially supported devices with built-in ANT support is maintained at <http://www.thisisant.com/directory/filter/~60/~/>.

To utilize ANT radio support, one or more service packages are also required to be installed on the phone (The service packages are available on the [Google Play Store](https://play.google.com/store/apps/details?id=com.thisisant.ant) and the ANT website at <http://www.thisisant.com/developer/resources/downloads/>):

2.1.1 ANT Radio Service

The ANT Radio Service manages controlling the ANT radio and sharing it between multiple apps. It is required to utilize ANT on all devices.

2.1.2 ANT USB Service

The ANT USB Service is required to communicate with ANT USB devices in USB host mode in addition to the ANT Radio Service. Note: not all Android devices support USB host mode.

2.2 ANT Libraries

For an application to use ANT it utilizes at least one of the provided ANT libraries (The libraries and their javadoc are available on the <http://www.thisisant.com> website):

2.2.1 PluginLib

The PluginLib enables applications to interact with the ANT+ Plugins. The PluginLib provides interfaces that are specialized to communicate with devices implementing the supported ANT+ Device Profiles. The PluginLib is the recommended way to interact with any ANT+ devices. See section 3 for more details.

2.2.2 *Android_AntLib*

The Android_ANTLib Library defines the API to use when not using the ANT+ profile plugins. See section 4 for more details.

2.3 ANT+ Membership and License Agreement

In addition to the services and libraries, in order to access various ANT related documentation, additional downloads, and receive support directly or through the ANT forums you must sign up for an ANT+ membership on <http://www.thisisant.com>. Signing up for an 'adopter' level account is free and provides access to most resources. Details on the various levels of membership are explained on the website.

3 Programming with the ANT+ Plugins

3.1 Overview of Using the ANT+ Plugins

The ANT+ Plugins are the recommended way to interact with ANT+ devices. Applications reference the antpluginlib.jar package in order to use the ANT+ Plugins. Interaction with a specific plugin type is done through the associated plugin communicator objects, called PCCs.

An application can utilize the plugins by following these general steps:

- 1) Request Access to the Plugin
 - Use the static method **requestAccess()** to secure access to the plugin for the type of device you want to communicate with.
 - The access request will return a PCC with an active connection to the plugin, or failure codes indicating problems accessing the plugin.
- 2) Subscribe to Events
 - Subscribe to events you wish to receive by calling the subscribe function on the PCC for the specific events you want to receive and passing your event receiver to handle the incoming data.
 - When you no longer wish to receive an event simply unsubscribe from it by passing null into the subscribe function for the event.
- 3) Use Device while Monitoring the Device Connection State
 - Plugins will inform apps of their state through the stateReceiver passed in the requestAccess function. The states reported can vary from device to device, but all devices will report fatal errors using the 'dead' state, which should always be handled by the app.
 - Some plugins also have request functions to send data to and control connected devices.
- 4) Release the PCC
 - When you are done with the device and are ready to revoke your plugin access call **releaseAccess()** on the PCC.

Please refer to the PluginLib javadoc for a full description of the APIs and to the ANT+ Plugin Sampler app for an example of how to use the API.

3.2 ANT+ Plugin Sampler application and source code

This ANT+ Plugin Sampler app (code provided as an Eclipse project) shows how to use the ANTPluginLib to connect to the ANT+ Plugins. Some of the sensors included in the sampler

are: heart rate monitor, stride based speed and distance monitor, and weight scale. It also shows how to handle device state changes and errors requesting access such as the scenario where the ANT Radio Service and/or ANT USB Service have not been installed yet.

4 Programming with the Android_ANTLib

4.1 Overview of Using the Android_AntLib

The Android_AntLib is used by applications when they are not using ANT+ Plugins. Typically an application would want to do this if it wants to directly control an ANT channel for non-ANT+ implementations.

Applications reference the android_antlib.jar package in order to use the ANT API.

An application can utilize the Android_AntLib by following these general steps:

- 1) Check for ANT support on the Device
- 2) Bind to the ANT Radio Service
- 3) Get the ANT Channel Provider
- 4) Acquire Channel(s)
- 5) Configure Channel(s)
- 6) Use Channel(s)
- 7) Release the Channel(s) and Unbind

Please refer to the Javadoc for a full description of the API and the Acquire Channels Sample app for an example of how to use the API.

4.2 ANT Acquire Channels Sample application and source code

This reference application (an Eclipse project) shows how to use the Android_ANTLib to acquire channels and use them to make simple connections to other devices. It also shows how to handle the scenario where the ANT Radio Service and/or ANT USB Service have not been installed yet.

5 Development Tools

5.1 Android SDK, Eclipse IDE and Android ADT

All Android application development requires the Android SDK.

The Eclipse IDE is the de-facto standard for Android development and is recommended for development. The ANT demo apps are packaged as Eclipse projects to get you up and running as quickly as possible.

The Android Development Tools (ADT) add-on is required for using Eclipse to develop Android applications.

The android website currently provides a single download with all the required files and the information to get started at <http://developer.android.com/sdk/index.html>

5.2 ANT Android Emulator Bridge

This ANT Android Emulator Bridge tool can be used with emulators running Android images that support ANT. The bridge allows an ANT USB stick that is connected to a PC running the bridge to be used by the emulator over the PC's loopback or Network interface. This allows ANT Android applications to be developed without the use of a physical android device. The emulator bridge tool is available for download on the Android API page in the developer's zone at thisisant.com.

<http://www.thisisant.com/developer/resources/downloads/>

The 'ANT Android Emulator Bridge Tool' downloadable package contains the bridge tool application along with documentation explaining how to use the tool itself and listing download locations for supported Android emulator images.

6 Reference and Support

6.1 ANT Message Protocol and Usage Document

The Andoird_AntLib API allows applications to control the ANT radio, interacting via the ANT Message Protocol. While there are some differences to the serial interface described therein, most of the Andoird_AntLib functions correspond to messages described in section 9.5 (ANT Message Details) of the ANT Message Protocol and Usage Document. This can be found in the Ant Developer's Zone at

<http://www.thisisant.com/resources/ant-message-protocol-and-usage/>

6.2 ANT+ Device Profile Documentation

The ANT+ Plugins allow devices to interact with devices implementing ANT+ Profiles at a high level of abstraction without worrying about most of the profile implementation and messaging details. However, a deeper understanding of the individual profile implementations and their use case details can help you to better understand how each plugin works and how to use them more effectively. The ANT+ Device Profile documents describe the detailed specifications of each profile and can be downloaded from the ANT Developer Downloads 'Documents' section:

http://www.thisisant.com/developer/resources/downloads/#documents_tab

6.3 ANT+ Forums

The ANT+ Forums provide a place to search for information, ask questions regarding ANT development, and find details on how to handle common issues you may encounter.

<http://www.thisisant.com/forum/>

6.4 Android Developers Guide

Android application development has several key differences when compared with developing for other operating systems. Knowledge of the Android framework and the Android application lifecycle will be essential to creating successful ANT+ Android applications. A good starting place is the Android "Getting Started" guide:

<http://developer.android.com/training/index.html>

7 Notes on Use of ANT+ Logos and Certification

The ANT+ logos are used to inform consumers of an application's interoperability with ANT+ devices. Only certified applications are allowed to use the ANT+ name, logos or icons.

Before using any ANT+ branding, the application must complete the ANT+ certification process to ensure that it complies with the device profiles it implements. The process is similar to the certification process for ANT+ sensors and devices but is streamlined for applications. This process is further streamlined for applications that make use of the ANT+ plugins. For more details please visit: <http://www.thisisant.com/developer/ant-plus/certification/> or contact certification@thisisant.com.

Once certification is complete the ANT+ logos can be used on both the application and promotional

materials such as on the Google Play Store or other websites where the application is available. Logo files will be distributed upon certification.