Getting Started with the Android Sample Application

Product Version 3.2.0.144

Published: 23-Jul-2013 09:57

Gracenote, Inc.
2000 Powell Street, Suite 1500
Emeryville, California
94608-1804
www.gracenote.com

Table of Contents

Introduction

The Mobile Client SDK provides a Sample Application that demonstrates basic functionality. The SDK also provides a development project that is an example of how to incorporate the Mobile Client into your Android application.
This document describes how to integrate the Sample Application project into your development environment.

Song Metadata Cache

Gracenote provides a local cache of fingerprints and metadata of some example songs. The SDK uses this data to attempt a local ID prior to attempting an online lookup. The SDK provides song samples in the `sample_music` folder that you can use to test local lookup identification. For additional samples, contact your Gracenote Professional Services representative.

Set Up Your Android Development Environment

The sample application requires an Android development environment. Set up the development environment by following the instructions here: http://developer.android.com/sdk/installing.html
When setting up the environment, be sure to install the following components, tools, and plug-ins. All of these are required for the Android Sample Application.

- Eclipse 3.5 (Galileo) or greater: http://www.eclipse.org/downloads/
- Eclipse JDT plugin, included in most Eclipse IDE packages: http://www.eclipse.org/jdt
- JDK 5 or JDK 6 (installing JRE alone is not sufficient): http://www.oracle.com/technetwork/java/javase/downloads/index.html
- Android Development Tools plugin for Eclipse: http://developer.android.com/sdk/eclipse-adt.html
- Android Software Development Kit: http://developer.android.com/sdk/index.html
- Google APIs Level 8, Revision 2 or later for Android. These are required for the Sample Application only, not for general Mobile Client development. See http://code.google.com/android/add-ons/google-apis/installing.html and http://developer.android.com/sdk/adding-components.html
  Android API version 2.2 or higher
- Android SDK tools directory in your PATH environment variable
- At least one Android platform installed
- At least one physical device installed

To create a suitable Android development environment for Mobile Client development:

1. Download and install the Android SDK from http://developer.android.com/sdk/index.html.

2. Download and install Eclipse IDE from http://www.eclipse.org/downloads/.

   ⚠ Be sure to download the Eclipse IDE version that is specified by the Android SDK.

3. Install ADT, the Android SDK plug-in for Eclipse. For instructions, see http://developer.android.com/sdk/eclipse-adt.html#installing.
   ADT allows Android platforms to be downloaded and installed, and Android virtual devices to be created and managed.
4. From the Preferences dialog box, configure the ADT plug-in with the location where the Android SDK is installed.
5. Install Android components via the ADT plug-in.
   Launch ADT from Eclipse and download and install the desired Android platform.

   ⚠ When developing on a Windows platform, be sure to download the USB drivers package. These are required for the development environment to communicate with a physical Android device.

Set Up Your Device

After setting up your Android development environment, you will need to set up one or more Android devices. For complete instructions, see the Android Developers Guide:

http://developer.android.com/guide/developing/device.html

When using a physical device, ensure that it has the following options checked:

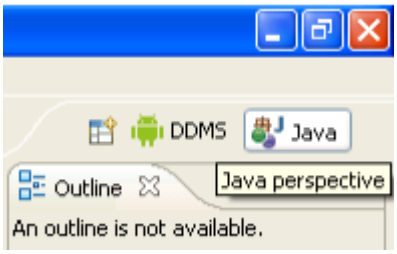
- **Unknown Sources** on the **Applications** menu
- **USB Debugging** on the **Development** menu

Create the Sample Application

Gracenote provides an Eclipse project for the Sample Application that you can import into your Android development environment. The project incorporates the Gracenote libraries and Sample Application source code.

1. Extract the files from the Mobile Client package to a location on your development machine. For example, extract the files to **C:\GN_Music_SDK_Android_x_x_x_x**, where "**_x_x_x_x**" is the current release number.
2. Write-enable the extracted folders (and files) so they can be updated by the build process.

3. Launch Eclipse and ensure you are in the Java perspective by clicking the Java perspective icon located in the top-right hand corner of the Eclipse page.



4. Select **File** > **New** > **Project** > **Android** > **Android Project from Existing Code**
5. In the **Import Projects** dialog box, **Browse** to, and select, the Mobile Client package location on your computer and click **Open**.
6. Click **Finish**. The project should now be created in your development environment.

> The above was done in Eclipse version Juno 4.2. In other Eclipse versions, the exact sequence and wording could be different.

Add Your Gracenote Client ID to the Sample Application Code

Before you can build and run the Sample Application, you must update the application to use your Client ID-Client ID Tag pair. Gracenote Professional Services provides both your Client ID and Client ID Tag.

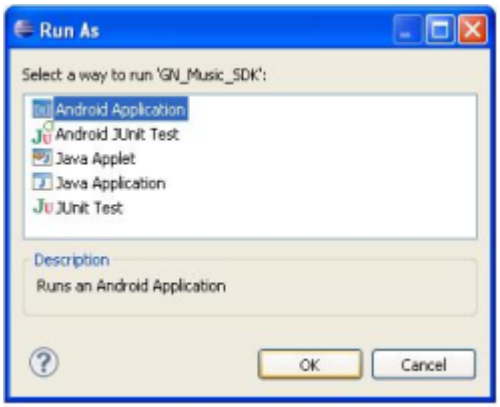
1. In Eclipse, open the Sample Application source file: **src/com/customer/example/GracenoteMusicID.java**
2. Locate `GNConfig.init()` in the `init()` method.
3. Add your Client ID-Client ID Tag pair as the first parameter to `GNConfig.init()`. A dash needs to separate the Client ID and Client ID Tag, e.g.: **123456-789123456789012312**.

Build and Run the Sample Application

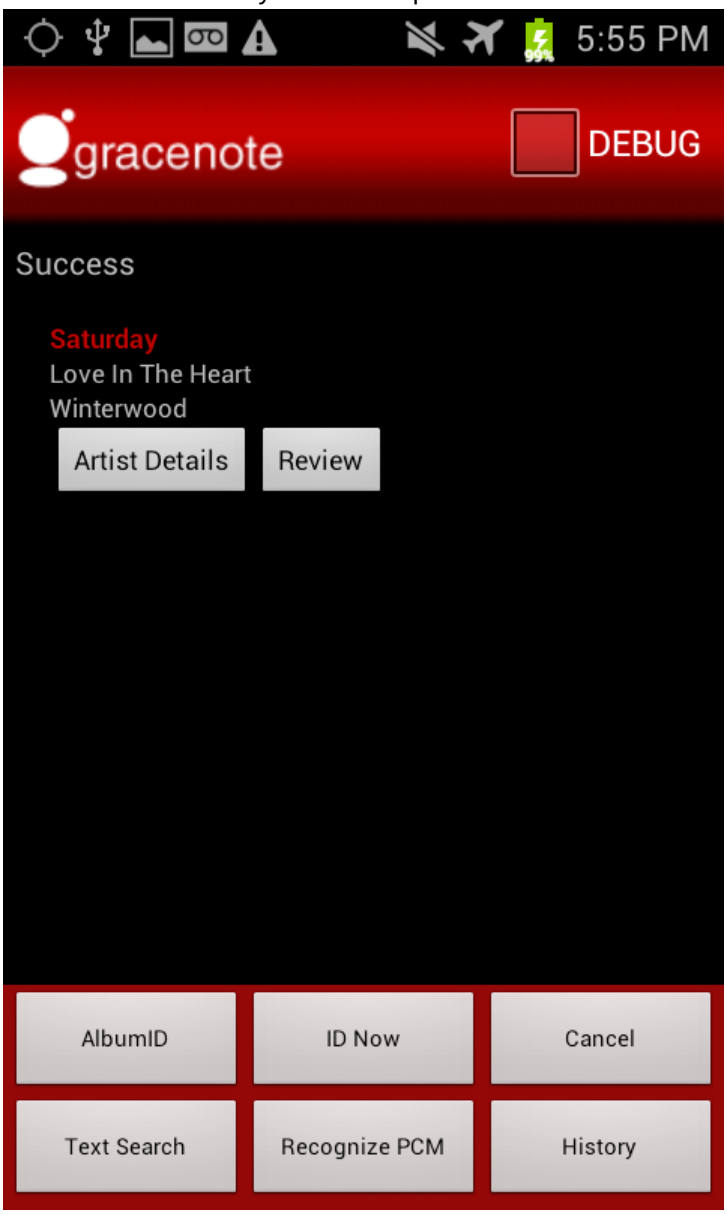
To build and run the Sample Application:

1. Make sure you have a connected device (hardware) to your development environment. If no devices are connected, Eclipse launches the first available virtual device.
    - For hardware devices:
        1. Connect the device to a USB port.
        2. Ensure that the device configuration enables the **Enable USB Debugging** and **Unknown Sources** options.
2. In Eclipse, choose **Run** > **Run**.

3. In the **Run As** dialog box, choose **Android Application** and click **OK**. This builds the application and deploys it to your connected device.



4. Confirm the application works: place your device near an audio music source and click the **id Now** button. If no results are returned, then the application could not identify the audio files. If this is the case, verify the audio is sufficiently loud and repeat the test. You should see a screen with results similar to this one:



The Sample Application contains code for the following operations, which have been omitted in the UI:

- Recognize files
- Lyrics search
- Recognize from mic (same functionality as ID-Now but slower and does not require a started continuous streaming session)

Troubleshooting

This section covers common problems with setting up or running the sample application.

"com.google cannot be resolved to a type"

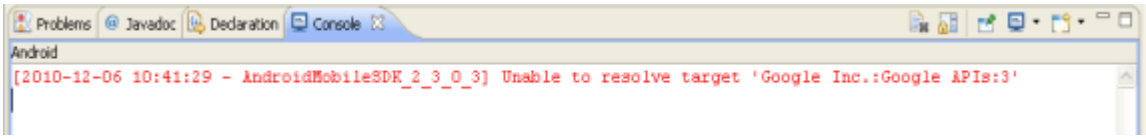
You need to reference the correct library at compile time. Right-click your Eclipse project, **Properties** > **Android** . In **Project Build Target** select **Google APIs 4.1**.

Issues Connecting to the Device on Windows

On Windows, you must install the ADB Composite ADB Interface USB driver. You can download this via the Android SDK and AVD Manager. For detailed instructions, see http://developer.android.com/guide/developing/device.html#setting-up.

Fix an Unresolved Reference to Google APIs

When the appropriate Google APIs are not installed into your Android Development Environment, then Eclipse generates an error message similar to the one shown below.

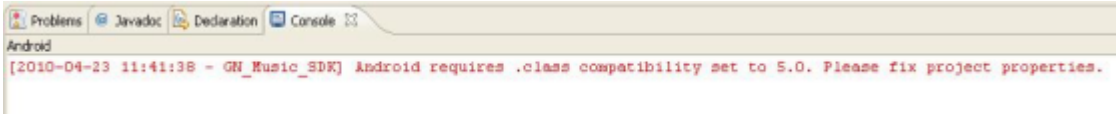


The Gracenote Mobile Client Sample Application uses Google APIs to determine the user's location when identifying a song; consequently the Google APIs must be included to correctly run the Sample Application project. Use the Android SDK and AVD Manager to add the appropriate Google APIs to your Android development environment.

Note: Google APIs are not required for Mobile Client; they are only required for the Mobile Client Sample Application.

Class Compatibility Error

When you build the Sample Application, Eclipse may generate a class compatibility error in the Console tab.



To fix this error:

1. Right-click on GN_Music_SDK_Android_<version number> in the **Package Explorer** tab.
2. Navigate to the **Android Tools** menu option
3. Select **Fix Project Properties** in the submenu.
4. Rebuild the project.