Creating and Using Databases for Android Applications

Sunguk Lee*

[1]*Research Institute of Industrial Science and Technology*
Pohang, Korea
sunguk@rist.re.kr
**Correspondent Author: Sunguk Lee* (sunguk@rist.re.kr)*

Abstract

This paper presents the overview of the Android platform towards software development for mobile and non-mobile applications. Android platform includes the popular open source SQLite database which has been used with great success as on-disk file format that allows the developer to handle data in a simple way, but also have the use of database features (such as undo, redo, etc.).

Keywords: *android, android platform, SQLite database*

1. Introduction

Android is an open source software assemble of an operating system, middleware and key applications for mobile devices introduced by Google capable of running multiple application programs. It is a complete operating environment based upon the Linux® V2.6 kernel. Initially, the deployment target for Android was the mobile-phone arena such as smart phones and low-cost flip-phone devices [1, 2].

Android platform is produced to make new and innovative mobile application program for the developers to make full use of all functions connected to handset internet. As a complete mobile platform, Android provides universal set of powerful Operation System, Comprehensive Library Set, abundant Multimedia User Interface and Phone Application. The data-storage burden is eased because the Android platform includes the popular open source SQLite database. SQLite is a software library that implements a SQL engine. It has been used with great success as on-disk file format: allows the developer to handle data in a simple way, but also have the use of database features (such as undo, redo, etc.). In embedded device environment, in which there is low-concurrency and there are little or medium size datasets, SQLite is the right choice.

In this paper, we discuss the overview of the android platform, it architecture and the android applications. The data storage for Android featuring the SQLite database was also discussed.

The rest of this paper is organized as follows: Section 2 explains the Android platform overview; Section 3 outlines the use of database for Android; Other databases for Android are identified in Section 4; and the concluding remarks in Section 5.

2. Overview of the Android Platform

Android is a set of software for mobile devices including Operation System, Middleware and Core Application, and a new Mobile Platform of Google. It is a complete mobile platform based on LINUX 2.6 Kernel that provides universal set of powerful Operation System, Comprehensive Library Set, abundant Multimedia User Interface and Phone Application.

Android platform is produced to make new and innovative mobile application program for the developers to make full use of all functions connected to handset internet. The Android platform was developed by Google and later the Open Handset Alliance (OHA).

The Open Handset Alliance™ is a group of mobile and technology companies who come together to accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience for consumers.

The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

Android is a multi-process system in which each application (and parts of the system) runs its own process. Security between applications and the system is enforced at the process level through standard Linux facilities such as user and group IDs that are assigned to applications.

Additional finer-grained security features are provided through a “permission” mechanism that enforces restrictions on the specific operations that a particular process can perform, and per-URI permissions for granting ad-hoc access to specific pieces of data.

2.1 Android Architecture

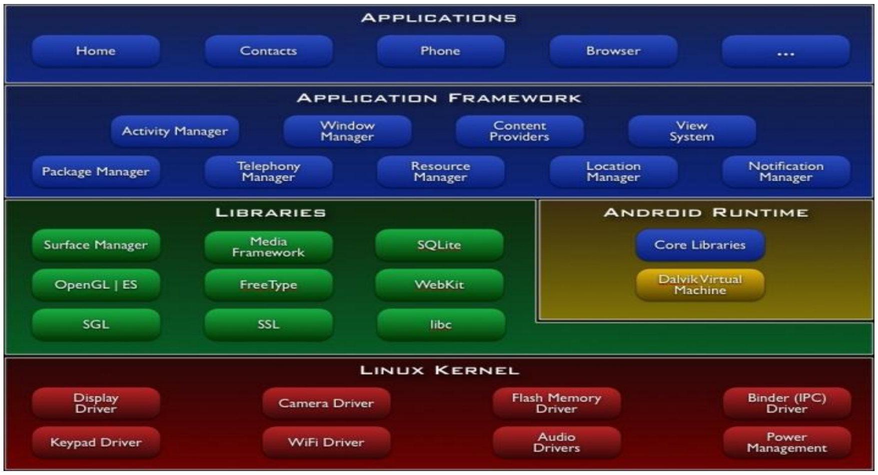


Figure 1. Android Architecture [5]

Linux Kernel. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack [5, 6].

Libraries. Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework.

- System C library - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices
- Media Libraries - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG

- Surface Manager - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications
- LibWebCore - a modern web browser engine which powers both the Android browser and an embeddable web view
- SGL - the underlying 2D graphics engine
- 3D libraries - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer
- FreeType - bitmap and vector font rendering
- SQLite – a powerful and lightweight relational database engine available to all applications

Android Runtime. Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint.

Application Framework. By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Applications. Android will ship with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others. All applications are written using the Java programming language.

2.2 Android Applications

Android runs atop a Linux kernel. Android applications are written in the Java programming language, and they run within a virtual machine (VM). Each Android application runs within an instance of the Dalvik VM (an open source technology), which in turn resides within a Linux-kernel managed process [10].

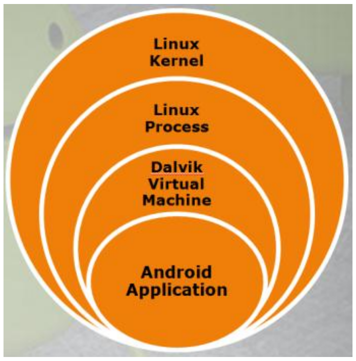


Figure 2. Android Application Architecture

An Android application consists of one or more of the following classifications [10]:

- Activities. An application that has a visible UI is implemented with an activity. When a user selects an application from the home screen or application launcher, an activity is started.
- Services. A service should be used for any application that needs to persist for a long time, such as a network monitor or update-checking application.
- Content providers. You can think of content providers as a database server. A content provider's job is to manage access to persisted data, such as a SQLite database. If your application is very simple, you might not necessarily create a content provider. If you're building a larger application, or one that makes data available to multiple activities or applications, a content provider is the means of accessing your data.
- Broadcast receivers. An Android application may be launched to process a element of data or respond to an event, such as the receipt of a text message.

An Android application along with a file called AndroidManifest.xml is deployed to a device. The AndroidManifest.xml contains the necessary configuration information properly installed to the device. It includes the required class names and types of events the application is able to process, and the required permissions the application needs to run. For example, if an application requires access to the network — to download a file, for example — this permission must be explicitly stated in the manifest file. Many applications may have this specific permission enabled. Such declarative security helps reduce the likelihood that a rogue application can cause damage on your device.

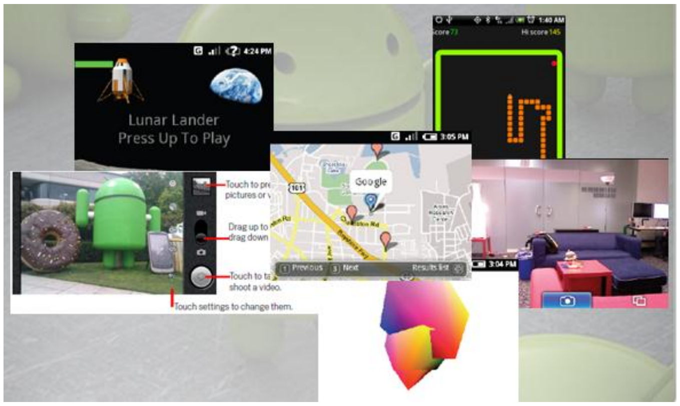


Figure 3. Sample Android Applications

3. Database for Android Application

The lifeline for every Android application is its database support. A database system is needed to store structured data, unless the application deals only with simple data. Android uses the SQLite database system, which is an open-source, stand-alone SQL database, widely used by many popular applications. SQLite is a lightweight transactional database engine that occupies a small amount of disk storage and memory, thus, it is a perfect choice for creating databases on many mobile operating systems such as Android and iOS.

The database that is created for an application is only accessible to itself; other applications will not be able to access it. Once created, the SQLite database is stored in the */data/data/<package_name>/databases* folder of an Android device [7].

3.1 What is SQLite

SQLite is an Open Source Database which is embedded into Android. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. In addition it requires only little memory at runtime (approx. 250 KByte). SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world [3, 7].

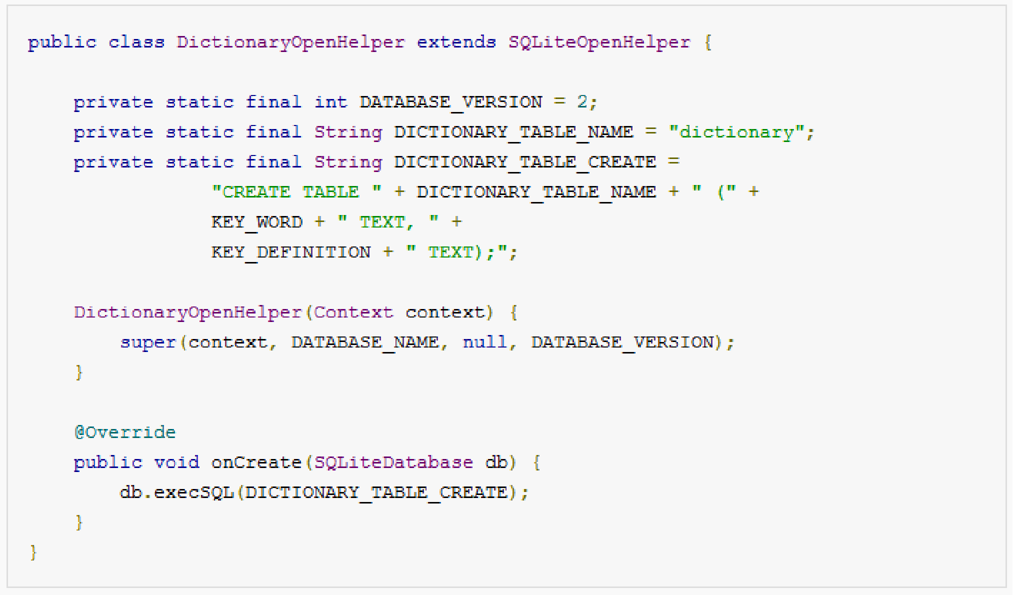
SQLite supports the data types TEXT (similar to String in Java), INTEGER (similar to long in Java) and REAL (similar to double in Java). All other types must be converted into one of these fields before saving them in the database. SQLite itself does not validate if the types written to the columns are actually of the defined type, e.g. writing an integer into a string column and vice versa.

SQLite is available on every Android device. Using an SQLite database in Android does not require any database setup or administration. The SQL statements for creating and updating the database must only be defined and the database is automatically managed for users by the Android platform.

3.2 Using the SQLite Database

Android provides full support for SQLite databases. Any databases that is created will be accessible by name to any class in the application, but not outside the application [4, 7].

The recommended method to create a new SQLite database is to create a subclass of *SQLiteOpenHelper* and override the *onCreate()* method, in which and SQLite command is executed to create tables in the database. An example is shown in Figure 4.



```
public class DictionaryOpenHelper extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 2;
    private static final String DICTIONARY_TABLE_NAME = "dictionary";
    private static final String DICTIONARY_TABLE_CREATE =
                "CREATE TABLE " + DICTIONARY_TABLE_NAME + " (" +
                KEY_WORD + " TEXT, " +
                KEY_DEFINITION + " TEXT);";

    DictionaryOpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(DICTIONARY_TABLE_CREATE);
    }
}
```

Figure 4. Sample Android Applications [7]

You can then get an instance of the *SQLiteOpenHelper* implementation using the constructor that is being defined. To write to and read from the database, the *getWritableDatabase()* and *getReadableDatabase()* are called, respectively. Both returns an *SQLiteDatabase* object that represents the database and provides methods for SQLite operations.

SQLite queries can be executed using the *SQLiteDatabase query()* methods, which accept various query parameters, such as the table to query, the projection, selection, columns, grouping, and others. For complex queries, such as those that require column aliases, the *SQLiteQueryBuilder* is used, which provides several convenient methods for building queries.

Every SQLite query will return a Cursor that points to all the rows found by the query. The Cursor is always the mechanism with which can navigate results from a database query and read rows and columns [7].

4. Other Databases

Cellica Database for Android allows viewing and updating the database contents on an Android device. It allows to sync with Microsoft Access, Microsoft Excel, and any ODBC Compliant database like Oracle, SQL Server etc. The software package consists of two software: the Cellica Database Desktop (Server) that runs on Microsoft Windows, and Android device Cellica Database (Client) that run on the Android device. The Desktop side software is used to create the database profiles (make the list of database) for Android phone/ Tablet and data will be synchronizing with Android phone/Tablet depending on the sync settings on phone/Tablet [8].

Cellica Database on Android phone/Tablet features include synchronized data wirelessly with Android phone/Tablet, view and update data, apply SQL Select queries, filters, sort the fields and synchronize data accordingly, supported password protection, supported forms, custom database creation, and so on.

Memento is a personal database that allows Android users to store all personal data in one place. Entries can be synchronized with Google Docs. For example, Memento handles collections, purchases, inventory, recipes, and tasks. Main features include storing entries with custom fields, sorting, grouping and filtering entries by any fields, password protection, synchronization with Google docs, sending entries via SMS, e-mail and other available devices, barcode scanning, backing up entries and restoring data, and so on [9].

5. Conclusions

Android is a software stack for mobile devices that includes an operating system, middleware, and key applications. The Android SDK provides the tools and libraries necessary to begin developing applications that run on Android-powered devices. Android provides full support for SQLite databases. Databases created will be accessible by name to any class in the application, but not outside the application.

The data-storage burden is eased because the Android platform includes the popular open source SQLite database. SQLite is a software library that implements a SQL engine. It has been used with great success as on-disk file format: allows the developer to handle data in a simple way, but also have the use of database features (such as undo, redo, etc.). In embedded device environment, in which there is low-concurrency and there are little or medium size datasets, SQLite is the right choice.

References

[1] http://en.wikipedia.org/wiki/Android_%28operating_system%29.

[2] http://en.wikipedia.org/wiki/Android_(operating_system).

[3] http://www.sqlite.org/.

[4] http://developer.android.com/guide/topics/data/data-storage.html#db.

[5] GOOGLE INC. (Hrsg.): Android Software Development Kit (SDK). Google Inc., http://developer.android.com/sdk/index.html. – Android 2.2, Release 2.

[6] http://elinux.org/Android_Architecture.

[7] http://developer.android.com/guide/topics/data/data-storage.html#db.

[8] http://www.cellica.com/cellicadatabaseandroid.html.

[9] https://play.google.com/store/apps/details?id=com.luckydroid.droidbase&hl=en.

[10] http://developer.android.com/guide/components/fundamentals.html.