INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

DESIGN AND BUILDING AN ANDROID APPLICATION BY USING INTEGRATED DEVELOPMENT ENVIRONMENT

ZEESHAN I. KHAN[1], VIJAY GULHANE[2]

1. ME, 1st Year , Sipna College of Engineering & Technology, Amravati.
2. Associate Professor, Dept. of COE, , Sipna College of Engineering & Technology, Amravati.

Accepted Date: 27/02/2014 ; Published Date: 01/05/2014

Abstract: A Smartphone is a device that lets you make telephone calls but also adds the features that you might find on a personal digital assistant or a computer. Today android is one of the most selling operating system in mobile and tablet devices forcing the developers to build an application for it. The paper gives the complete procedure for building an android application by using the graphical interface of desktop developing applications like Eclipse, Net beans, etc. The topic also explores the architecture of android operating system, Dalvik Virtual Machine as well as the 5 steps which are needed to build an application for android dumped gadgets the steps for making a live demo application.

Keywords: Kernel, ADT, IDE, Dalvik, Virtual Machine, Framework, SDK, Package.

Corresponding Author: MS. ZEESHAN I. KHAN

Access Online On:

www.ijpret.com

How to Cite This Article:

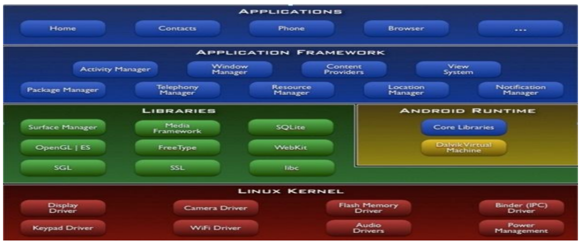
Zeeshan Khan, IJPRET, 2014; Volume 2 (9): 200-205

PAPER-QR CODE

INTRODUCTION

Android is a software platform and operating system for mobile devices, based on the Linux kernel, and developed by Google and later the Open Handset Alliance. Open Handset Alliance, is a consortium of several companies which include Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Sprint Nextel and NVIDIA, etc. There are currently more than 250,000 apps available for Android. Android Market is the online app store run by Google, though apps can also be downloaded from third-party sites. Developers write primarily in the Java programming language, controlling the device via Google-developed Java libraries.

Architecture:



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.

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.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework).

This same mechanism allows components to be replaced by the user.

Underlying all applications is a set of services and systems, including:

A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser.

Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data.

A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files.

A Notification Manager that enables all applications to display custom alerts in the status bar

An Activity Manager that manages the lifecycle of applications and provides a common navigation back stack.

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. Some of the core libraries are listed below:

- 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'sOpenCORE; 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. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

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.

Application Building:

Installation and Configuration of Supporting Tools like

Java JDK: Lays the foundation for the Android SDK.

Android SDK: Provides access to Android libraries and allows you to develop for Android.

Eclipse IDE (integrated development environment): Brings together Java, the Android SDK, and the Android ADT (Android Development Tools), and provides tools for you to write your Android programs.

Android ADT: Does a lot of the grunt work for you, such as creating the files and structure required for an Android app.

Step1. Set up Android Virtual Devices or hardware devices. You need to create Android Virtual Devices (AVD) or connect hardware devices on which you will install your applications.

Step2. Create an Android project

If you are using Eclipse, builds are generated each time you save changes and you can install your application on a device by clicking Run. If you're using another IDE, you can build your project using Ant and install it on a device using adb.

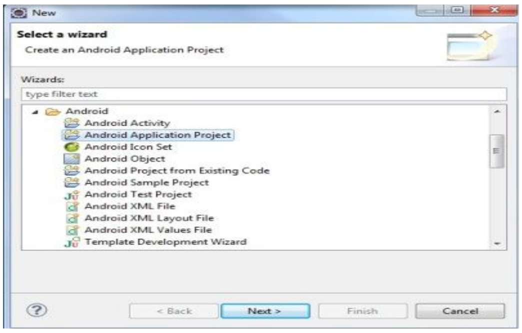
Step3. Debug your application with the SDK debugging and logging tools.

Step4. Debugging your application involves using a JDWP-compliant debugger along with the debugging and logging tools that are provided with the Android SDK. Eclipse already comes packaged with a compatible debugger.

Step5. Test your application with the Testing and Instrumentation framework. The Android SDK provides a testing and instrumentation framework to help you set up and run tests within an emulator or device.

Sample Application formation

a) From Eclipse, Select File > New > Android Application Project



New

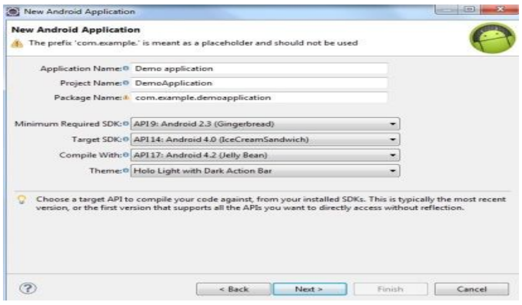
Select a wizard

Create an Android Application Project

Wizards: type filter text

- Android
  - Android Activity
  - Android Application Project
  - Android Icon Set
  - Android Object
  - Android Project from Existing Code
  - Android Sample Project
  - Android Test Project
  - Android XML File
  - Android XML Layout File
  - Android XML Values File
  - Template Development Wizard

b) Give Application name, Project name and Package name



New Android Application

New Android Application

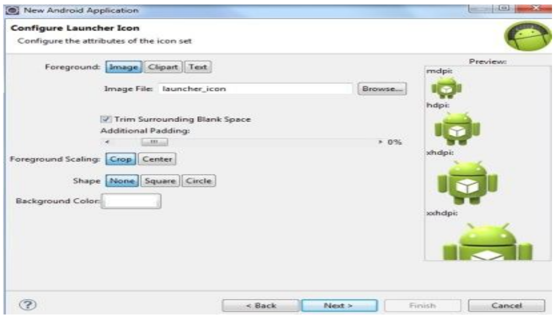
The prefix 'com.example.' is meant as a placeholder and should not be used

Application Name: Demo application
Project Name: DemoApplication
Package Name: com.example.demoapplication

Minimum Required SDK: API 9: Android 2.3 (Gingerbread)
Target SDK: API 14: Android 4.0 (IceCreamSandwich)
Compile With: API 17: Android 4.2 (Jelly Bean)
Theme: Holo Light with Dark Action Bar

Choose a target API to compile your code against, from your installed SDKs. This is typically the most recent version, or the first version that supports all the APIs you want to directly access without reflection.

c) Configure the icon properties



New Android Application

Configure Launcher Icon

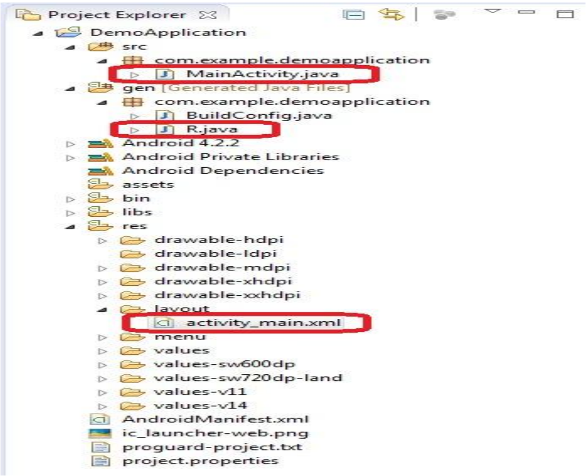
Configure the attributes of the icon set

Foreground: Image Clipart Text
Image File: launcher_icon
☑ Trim Surrounding Blank Space
Additional Padding: 0%
Foreground Scaling: Crop Center
Shape None Square Circle
Background Color:

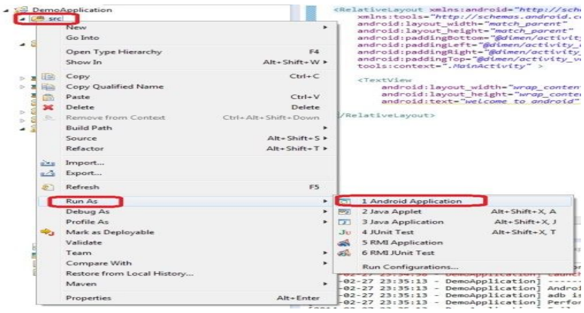
d) Main Activity.java helps you to write the java code in it.

Activity_main.xml is used for designing the application in the form of tags.

R.java is the main file maintained by the system itself to store the id of all the contents used in the application.



e) Launching the application



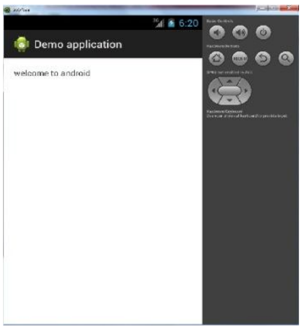
```
<RelativeLayout xmlns:android="http://sch
    xmlns:tools="http://schemas.android.c
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activit
    android:paddingLeft="@dimen/activity_
    android:paddingRight="@dimen/activity_
    android:paddingTop="@dimen/activity_v
    tools:context=".MainActivity" >

    <TextView
        android:layout_width="wrap_conten
        android:layout_height="wrap_conte
        android:text="welcome to android"

/RelativeLayout>
```

```
-02-27 23:35:13 - DemoApplication] ------
-02-27 23:35:13 - DemoApplication] Androi
-02-27 23:35:13 - DemoApplication] adb i
-02-27 23:35:13 - DemoApplication] Perfor
```

f) Running Application in Android Virtual Device



REFERENCES:

1. http://developer.android.com/guide/components/fundamentals.html

2. http://www.android-app-market.com/android-architecture.html

3. Android Application Development for Dummies.