


Java to apk

 I'm not robot  reCAPTCHA

Continue

Java Development Kit (JDK)





[Java to apk download.](#) [Javascript to apk.](#) [Java file to apk.](#) [Javascript to android .apk.](#) [Java code to apk.](#) [Java to apk online.](#) [Java compile to apk.](#) [Java to apk games.](#)

Bint Harwani, author of *Android Programming Unleashed*, helps Java programmers understand the main differences between Java and Android applications and the actions needed to convert or return Java as an Android program. Do you like this article? We recommend finding the code in Listing 1 of a simple Java program called `WelcomeMsg.java` that prompts the user for a name. After entering the name, the user will press the welcome message including the user entered by the user. For example, if the user types in the name "Kelly," the message "Hello, Kelly!" appears on the screen. 1 `WelcomeMsg.java` list code. `import java.io.*; #1 class Welcommsg { public static void main(String args[]) { InputStreamReader istream = new InputStreamReader(System.in); #2 buffereDreader buffRead = new BufferedReader(istream); #3 try { #4 system.out.println("Enter your name:"); #5 line name = buffread.readline(); #6 System.out.println("Hello" + Name + "!"); #7} catch (IOException err) { #8 system.out.println("Error reading string"); }}` `WelcomeMsg.java` set #1 imports the input/output data needed for the operation. Command 2 creates a scanner named `istream`. The reader reads from the standard input stream (`System.in`); This is from the keyboard. The entered text is preserved in the form of Unicode characters. 3 The instruction uses the ITRAM scanner to convert the data into a line shape. 4 The statement defines an attempt to verify that there are no errors that may have occurred while the reader was entering the text. Instructions 5 and 6 prompt the user for a name, which is then assigned to the line name. The entered name will then appear on the screen according to instruction No. 8. After starting the program, the user will be prompted to enter the name. Once the user has typed and typed, a welcome message will appear with the entered name as shown in Figure 1. Figure 1. Android Project Java release, like Java, is now developing the same Android application. Assuming you have the Java Development Kit (JDK) installed on your computer, download Android Studio. Then do the following: To install Android Studio, duplicate the downloaded executable twice. After successful installation, launch Android Studio. The first screen that opens is the splash screen which shows icons like new project, import project, open project and so on. Click the new project icon to create a new Android project. In NewWin the dialog box, enter the information for the new project as shown in Figure 2. In the Application Name area, enter the name of the Android project. Name the request `WelcomeMsg`. The module name entry is assigned automatically; By default, this is the same as the application name. To make it a unique identifier, the name of the attributed package is `com.yourname.welcommsg`. The location for the Android project is specified in the project location area. Select API 8: Android 2.2 (Froyo) from the required SDK drop-down list to indicate that the app requires at least API level 8. Select API 17: Android 4.2 (Jelly Bean) as your target platform as we assume this is the version most commonly used by your target audience. Select API 17: Android 4.2 (Jelly Bean) as the platform to compile the project. The default theme dedicated to the project is Holo Light with a dark action bar. By default, a custom build icon is created. This feature allows you to customize the application launch icon. Since we want to create an empty activity, click on the "Create activities" box and then click on the "Next" button. Figure 2 Creating a new Android project. The following dialog box is the launch icon configuration used to customize the application icon. Since we want to use the default icon for our application, leave the default options selected in the dialog box, and then click Next. The next dialog asks us to create an activity. Since we want to create an empty activity, select the "Empty Activity" option from the list and then click "Next". In the next dialog you will be asked to enter information about the newly created activity. Name the activity `WelcomeMsgActivity`. The default name `Activity_main.xml` is assigned to the layout file. Keep this default layout file name and click Finish to create the Android project. Android Studio will automatically create some files and folders for our new Android project (see Figure 3). To set up our new project, we only need to work with two files: `Activity_main.xml`. This file is shown on the right side of Figure 3 under the `WelcomeMsg> SRC> Main> Res> Mayout` node. This is the XML file where we will define the controls (`TextView`, `EditText`, `Button`, etc.) for our Android application's GUI. Through this file, the user interacts with the Android project. `welcommsgactivity.java`. This file is located in the node `WelcomeMsg > SRC > Main > Java > com.yourname.welcommsg`. This java file is loaded by the controls defined in `Activity_main.xml` listens for various events and runs the desired code if any of those events occur. Figure 3 Windows Project Explorer showing the files and folders of our new Android design in Twisted Mode (left) and unpackaged (right) mode control in the system container. The most popular way to create an Android UI is with an XML file. The control or views defined in the XML file for use in the Android application are or are located in the system container. Android provides several system containers: `RELATIVELAYOUT`, `LINEARLAYOUT`, `GROUOUT`, `FRAMELAYOUT` and so on. The default layout applies. The simplest is `LinearLayout`, which displays the controls linearly, below. Questionnaire Then replace the default content with the code specified in List 2. List 2 code for `Activity_main.xml`. Android > AVD Manager. The Android Virtual Device Manager dialog opens and displays a list of available AVDs. You can use this dialog to work with an existing AVD or to create a new AVD. Click NEW to create a new AVD. The dialog box creates a new Android (AVD) virtual device. Complete the dialog box as shown in Figure 5. Title AVD. Set the name to AVD. In this example, use the name `Phoneavd`. Apparatus. Enter the device you want to test the app on. In this example, select 5.1 WVGA (480 x 800: MDPI). Target. Enter the target API level. Our application will be tested against the specified API, so we define the target in the last API, Android 4.2.2 - API level 17. CPU/ABI. This represents the processor we want to emulate on our device. Select ARM (Armeabi-V7a). Keyboard. If you want to use the PC keyboard displayed on the emulator screen, check if the keyboard field is available. Skin. Check out the hardware widgets showcase. The emulator appears with physical device buttons on the right side. The controls include basic elements such as speakers and stop/stop buttons, accept, menu, back and search buttons. Front/rear camera. If you have a webcam connected to your computer and want to use it in your application, select WebCam0 from the drop-down. If you don't have a webcam, select the Emulated option. If the app does not require a camera, leave the default value. Memory options. Define RAM device and virtual memory segment parameters. Leave the default values here. Inner memory. Define internal device storage. Leave the default (200 MIO). SD card. Increases the storage capacity of the device. BigFiles such as audio and video (for which the built-in flash memory is not sufficient) are saved on the SD card. Let's set the SD card size to 128 MB. The more storage space is assigned on the SD card, the longer the creation of the AVD. If it is really necessary, keep the storage space on your SD card as low as possible. Snapshot. Activate this option to avoid starting the emulator and start with the recently stored snapshot. This option is used to start the Android emulator quickly. Use the main GPU. This option activates the GPU emulation, which improves the performance of the emulator. Rice. 5 Create a new virtual Android device (left). The newly created AVD is displayed in the Android Virtual Device Manager (right). After entering the data, click OK. A newly created AVD, `PhoneAVD`, is created and displayed in the list of existing AVDs, as shown on the right in Figure 2. 5. Click the Update button if the new AVD does not appear immediately in the list. Our new app is ready to go! Click on the "Run" symbol in the toolbar, press Shift-F10 or select "Execute"> "Run". You will be asked to select the device or AVD on which the output is displayed. Select `Phoneavd` from the dropdown list for virtual Android devices and then click OK. The Android project will ask for a name. After you have entered the name, click the Click ME button and a new welcome message is displayed (see Figure 6). Figure 6. Select a device or AVD to display the project edition (left) and enter a name if you are requested. Your new Android project shows a summary of the welcome message (right). You have learned how easy it is to convert every Java application into an Android application. Simply define the user interface in XML; Then use the Java codes stored in the activity file to access the data entered in the UI control. If you want to learn the Android programming step by step, read my book *Android Programming Unleashed* and discover the functions of this amazing smartphone platform. Platform.

