

DRAFT COPY: CONTENTS SUBJECT TO CHANGE

FTC Training Manual

Running App Inventor Locally for Windows PCs

Thomas Eng

9/10/2015



This document contains training material to introduce students and mentors to the next gen FIRST Tech Challenge robot controller platform.

Contents

1	Introduction	2
2	Installing App Inventor Locally on a Windows Computer.....	3
2.1	Installing Google Chrome.....	3
2.2	Downloading VirtualBox	3
2.3	Running the VirtualBox Installer	4
2.4	Verifying Custom Setup Features.....	4
2.5	Verify Custom Setup Options.....	5
2.6	Network Interfaces Warning.....	6
2.7	Ready to Install.....	6
2.8	Installing Device Software.....	7
2.9	Installation Complete!	7
2.10	Configuring VirtualBox	8
2.10.1	Configuring the Network Preferences	8
2.10.2	Verify the Network Settings.....	9
2.11	Installing the Appliance.....	11
2.12	Start the Appliance	14
2.13	Accessing the App Inventor from your Local Computer	16
2.14	Shutting Down the App Inventor Appliance	18
3	Let's Build an App!	19
3.1	Navigating the Designer Mode Screen	19
3.2	Creating Our User Interface.....	24
3.3	Programming in Block Mode.....	28
3.4	Building Your App.....	35
3.4.1	A Note Regarding App Inventor Local and Using a QR Code	35
3.4.2	Using a USB Cable to Install Your App.....	36

1 Introduction

The App Inventor is a design tool that makes it easy for you to create your own Android apps using a very user-friendly drag-and-drop, visual interface. In order to customize the behavior for your robot, you will need to use the App Inventor to create a Robot Controller app with one or more op modes that you can use to control your robot.

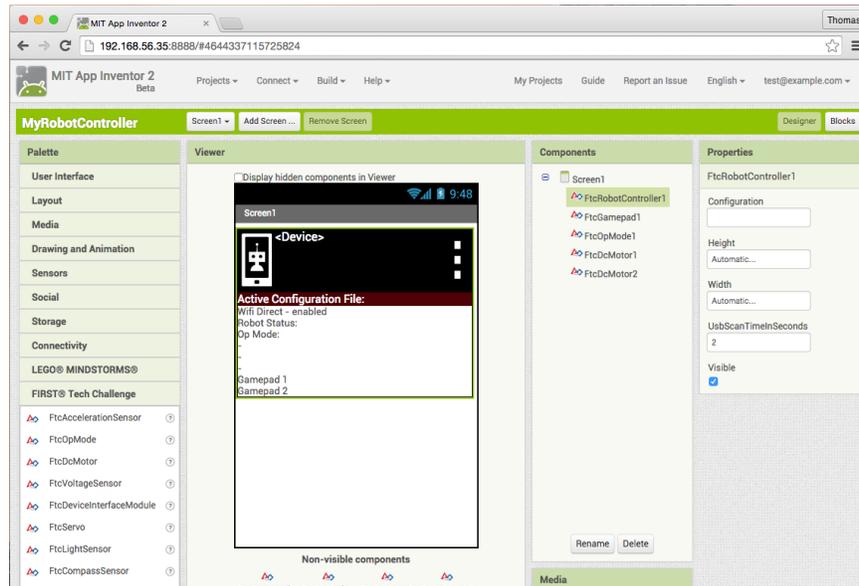


Figure 1 - The App Inventor is a browser-based application that makes it easy to create custom Android apps.

The App Inventor is a browser-based application. This means that you run the application using a web browser such as Google Chrome, Apple Safari, or Mozilla Firefox. The App Inventor is normally available as a web site run by MIT. This means that you typically need to be connected to the Internet in order to use App Inventor.

However, for the *FIRST* Tech Challenge (FTC) competitions teams might not always have access to the Internet at competition venues. This document teaches you how to install and run App Inventor for FTC on a computer so that it can be used *offline* without an Internet connection.

The App Inventor application will run in an *Appliance*, which is another way of saying that it will run in a *Virtual Machine*. The appliance is a virtual server running on your computer. This virtual server is preconfigured with all of the software that is needed to run App Inventor. Your personal computer will “talk” to this virtual server through a virtual (“host only”) network connection. Your personal computer will access the App Inventor application through this virtual server.

Important Note: Although the App Inventor virtual server can run locally without an Internet connection, you will need an Internet connection (or have the software copied on a flash drive) to install and configure the local copy of App Inventor.

2 Installing App Inventor Locally on a Windows Computer

In order to run the App Inventor locally, you will need to install a software package called VirtualBox onto your personal computer. VirtualBox is software that is available for free from a company called Oracle. It allows you to run virtual computers/servers on your personal computer. Once you have VirtualBox properly installed and configured, you will be able to run a virtual Linux server that has App Inventor built in.

2.1 Installing Google Chrome

Before you install VirtualBox onto your Windows computer, you should first make sure that you have a compatible web browser on your computer. Unfortunately, we do not recommend the use of Microsoft Internet Explorer with the App Inventor. Instead, we recommend that you install Google Chrome and use it as your browser on your Windows machine.

The Chrome web browser is available for free from Google. Visit the Chrome website (<http://www.google.com/chrome>) and follow the site's instructions to download and install Google Chrome onto your Windows PC.

2.2 Downloading VirtualBox

After you have installed Google Chrome onto your computer, you will need to install the VirtualBox software. VirtualBox lets you run a self-contained virtual machine within your personal computer. VirtualBox is available for free from Oracle.

Visit the following link to download the VirtualBox software:

<https://www.virtualbox.org/wiki/Downloads>

You will want to download the “VirtualBox ... for Windows hosts” package.



Figure 2 - Download VirtualBox for Windows hosts.

2.3 Running the VirtualBox Installer

Once VirtualBox has downloaded, navigate to the folder where it was downloaded (typically in the Downloads folder), and find the .exe file that was downloaded. Then, right-click on the file, and select **"Run as Administrator"**. If asked if you want to **"allow the following program to make changes to this computer"**, click **Yes**. Note: You need to have administrative privileges in order to do this step. If you are using a personal computer, you likely have administrative privileges. If you are using a school computer, you will probably not have administrative privileges, and should contact an administrator for assistance.

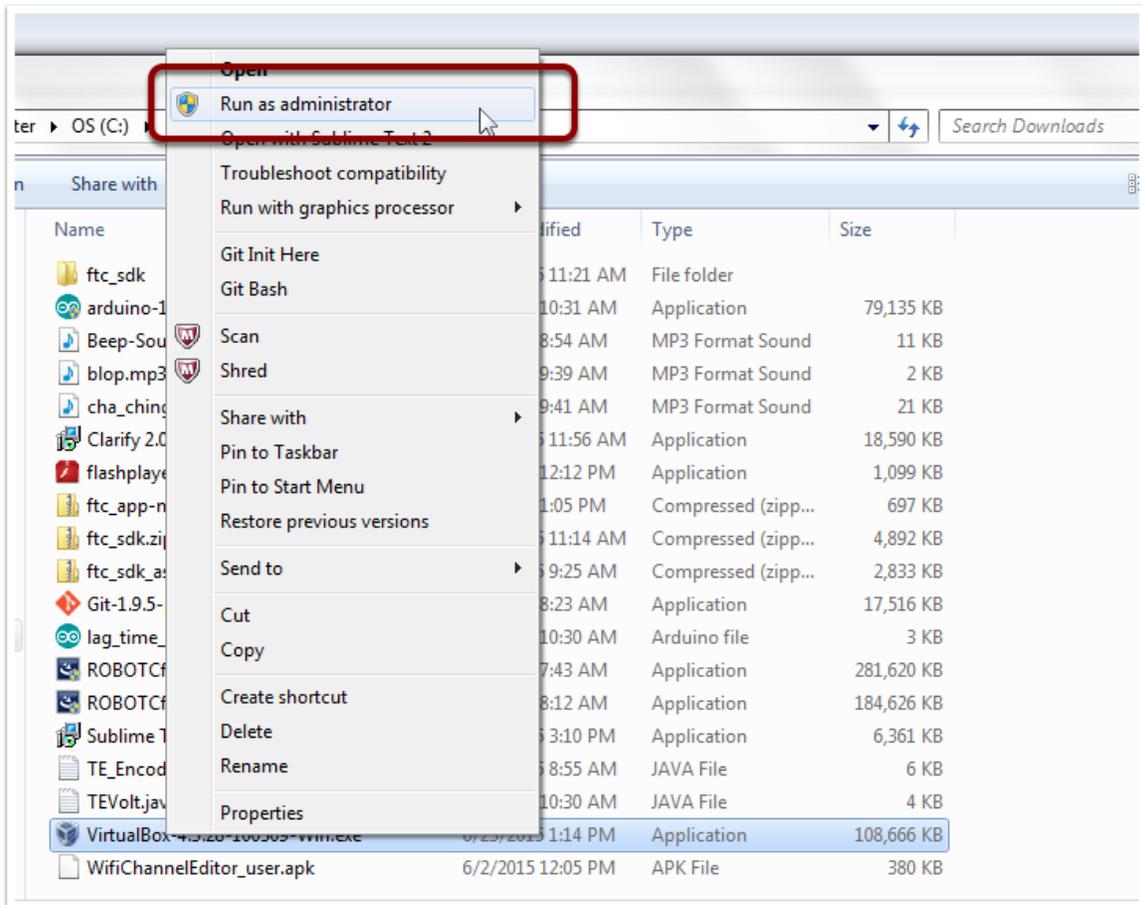


Figure 3 - Locate the VirtualBox software and run it as an administrator.

2.4 Verifying Custom Setup Features

Once the window for the **VirtualBox Installer Wizard** appears, click **Next**. When the **Custom Setup** features appear, make sure that all of the features are going to be installed for VirtualBox. If your options are slightly different than shown below, just make sure that all of the options are selected, and that **none of the options in the tree have a red X**. Then click **Next**.

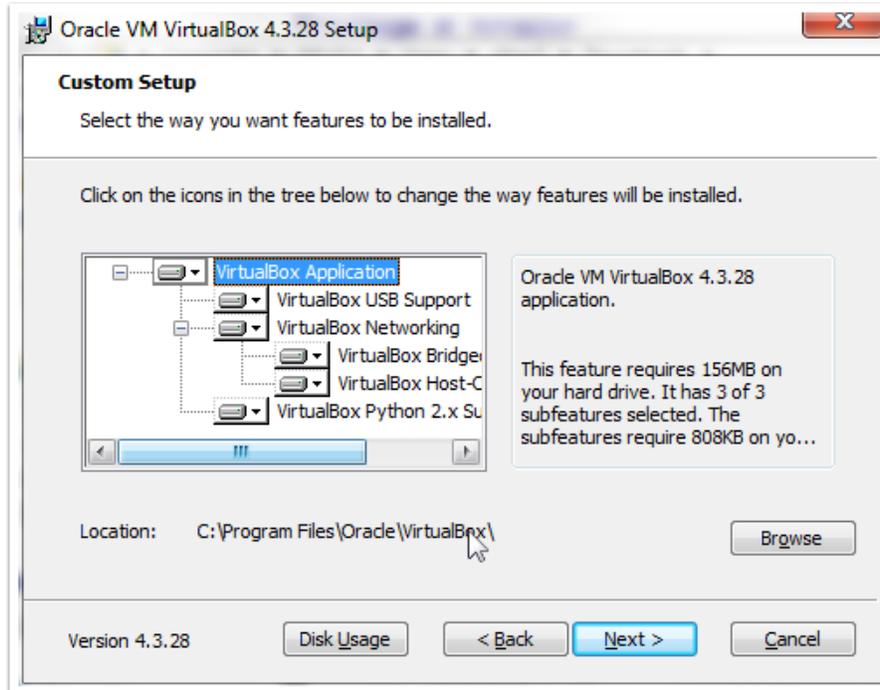


Figure 4 - Verify that your Custom Setup Options look like the ones depicted here.

2.5 Verify Custom Setup Options

When the window for **Custom Setup** appears, it is recommended that at least the "**Register file associations**" box is checked. If you would like a shortcut for VirtualBox on your desktop, check the "**Create a shortcut on the desktop**" box. If you would like a shortcut for VirtualBox in the toolbar next to the Windows icon, check the "**Create a shortcut in the Quick Launch Bar**" box.

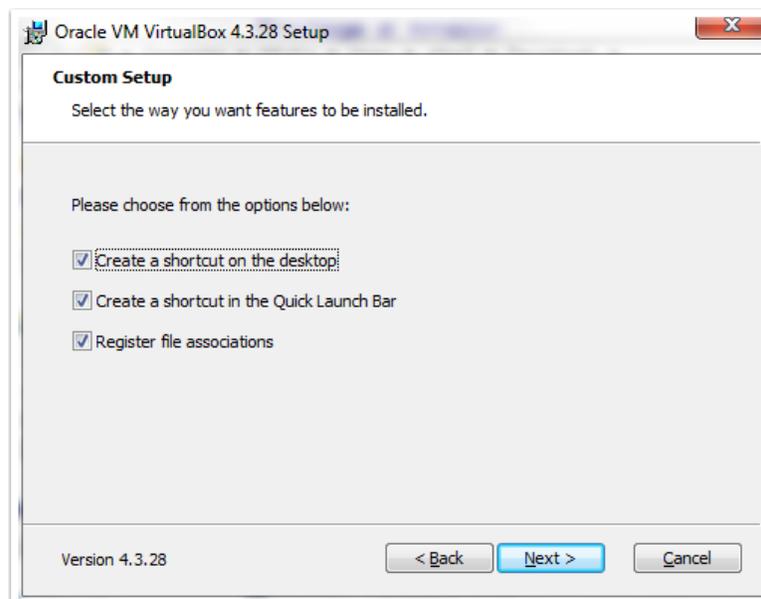


Figure 5 - Verify that the "Register file associations" option is selected and hit Next.

2.6 Network Interfaces Warning

A window labeled "**Warning: Network Interfaces**" should appear. The window will have a warning that the install process might reset your network connection. Don't worry – you can ignore this message and just click **Next**.



2.7 Ready to Install

When the window with "**Ready to Install**" appears, click **Install**.

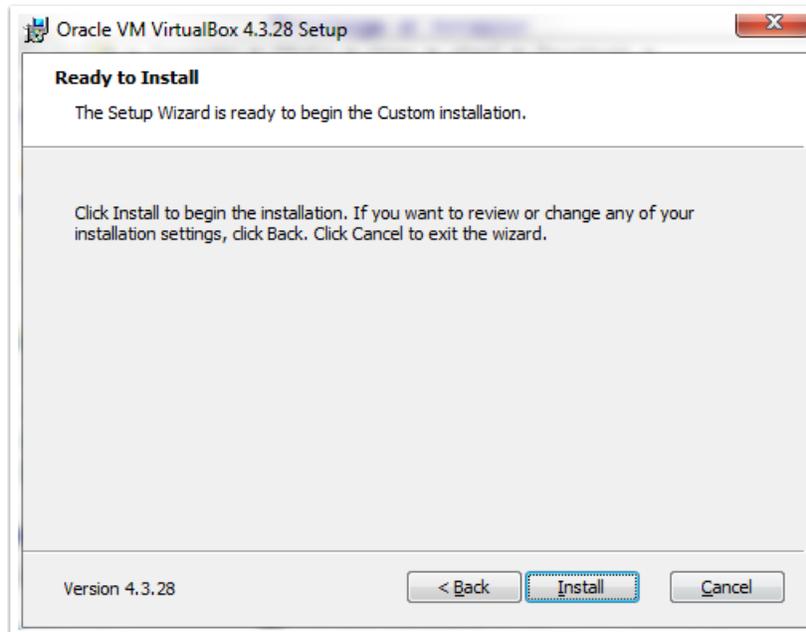


Figure 6 - Click on Install to continue with the installation.

2.8 Installing Device Software

At this point in the install process, your computer might prompt you if you "**would like to install this device software**" - click **Install** each time you are prompted.

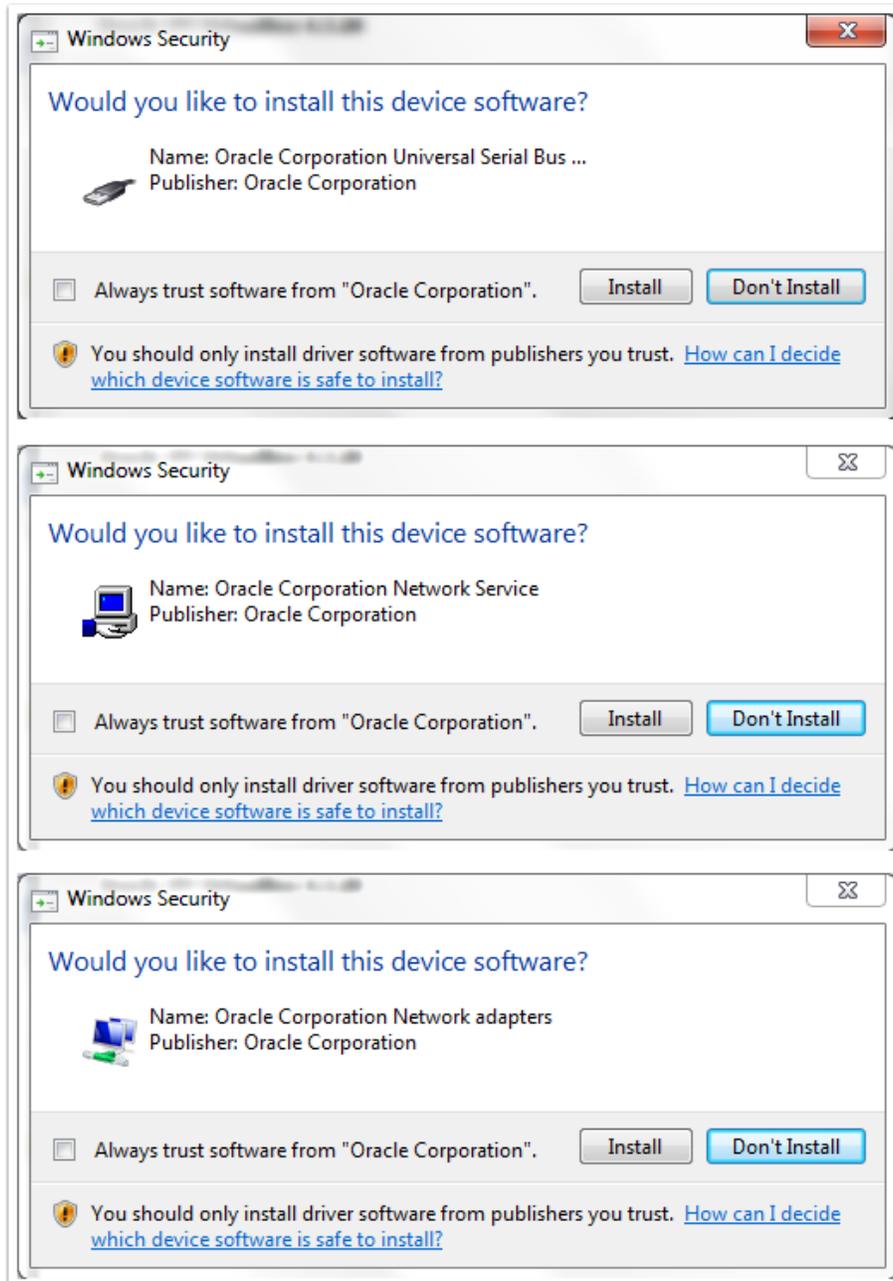


Figure 7 - Your computer might prompt you several times to install device software. Click Install for each prompt.

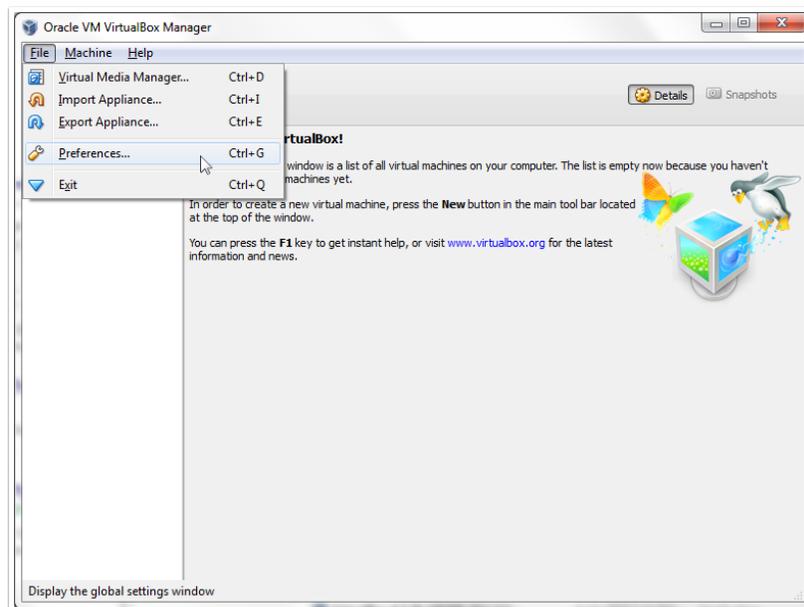
2.9 Installation Complete!

After the installation is finished (which may take a few minutes), a window with "**Installation is complete**" will appear. Leave the checkbox for "**Start Oracle VM Virtualbox after Installation**" checked, and click **Finish**.



2.10 Configuring VirtualBox

Now that we have installed the VirtualBox software, we need to configure it. When the "Oracle VM VirtualBox Manager" appears, click **File -> Preferences**.



2.10.1 Configuring the Network Preferences

In the **Preferences** menu, click the **Network** option.

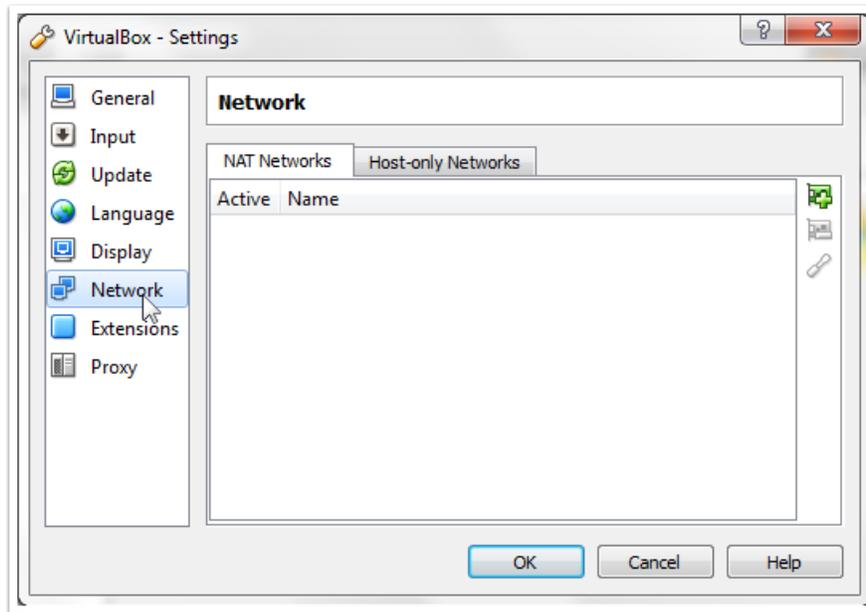


Figure 8 – Click on the Network option to display the network settings.

Click on the "**Host-only Networks**" tab, and make sure that the **VirtualBox Host-Only Ethernet Adapter** appears. If it does not, click the "**Add host-only network**" button, and one should appear. If the computer prompts you to "**allow changes**", click **Yes**.

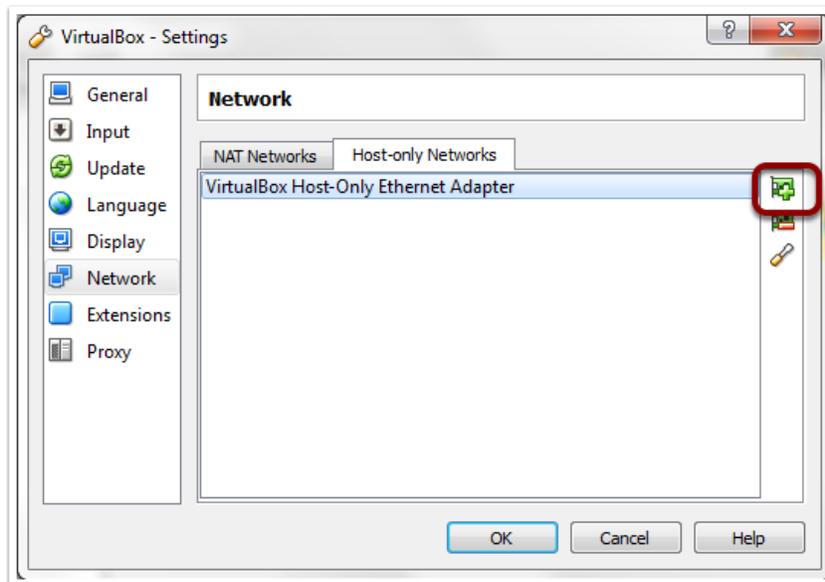


Figure 9 - Verify that a "VirtualBox Host-Only Ethernet Adapter" is present. If not, add one.

2.10.2 Verify the Network Settings

Click the "**VirtualBox Host-Only Ethernet Adapter**" to highlight it, and then click the edit button.

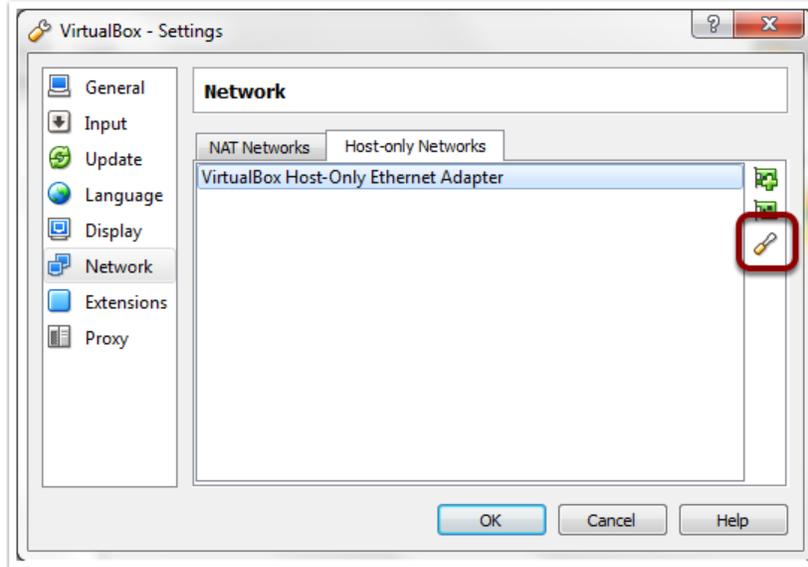


Figure 10 - Click on the Edit button to view the Host-Only settings.

Click the **Adapter** tab of the **Host-only Network Details** window, and make sure that the text fields are as follows:

IPv4 Address: 192.168.56.1

IPv4 Network Mask: 255.255.255.0

IPv6 Address:

IPv6 Network Mask:

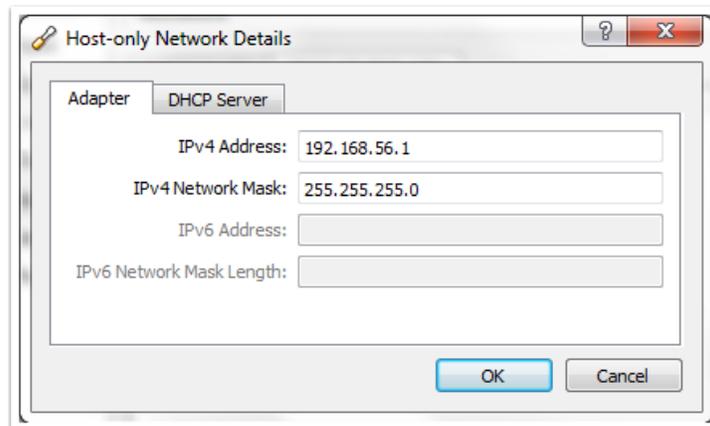


Figure 11 - Verify the Adapter settings.

Then, click the **DHCP Server** tab, and make sure the settings are as follows:

Enable Server: checked

DRAFT: Contents Subject to Change

ServerAddress: 192.168.56.100

Server Mask: 255.255.255.0

Lower Address Bound: 192.168.56.101

Upper Address Bound: 192.168.56.254

Then click **Ok**.

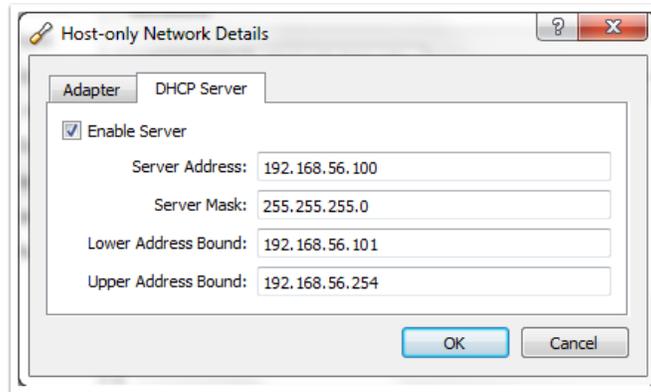


Figure 12 - Verify the DHCP Server settings.

Now click **Ok** to get back to the **VirtualBox Manager**.

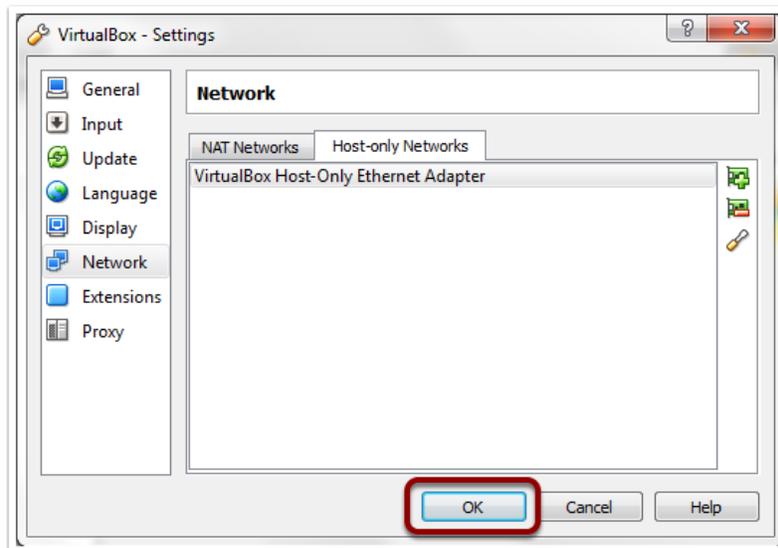


Figure 13 - Click on OK to return to the VirtualBox Manager.

2.11 Installing the Appliance

Now that we have the VirtualBox software configured, we need to download the *appliance* file. This file contains an “image” of a server that is configured to run the App Inventor application.

DRAFT: Contents Subject to Change

The appliance file is the file that contains the *image* of the virtual App Inventor server. The image file is available on an FTC App Inventor download page. You can access the FTC App Inventor download page using the following link (click on the button labeled "App Inventor (Windows)"):

<http://frc-events.usfirst.org/2015/ftcimages>

Also note that this file is large (approximately 2 GB). Depending on the speed of your Internet connection, it might take a while to download.

Once the file has downloaded, click **File -> Import Appliance**.

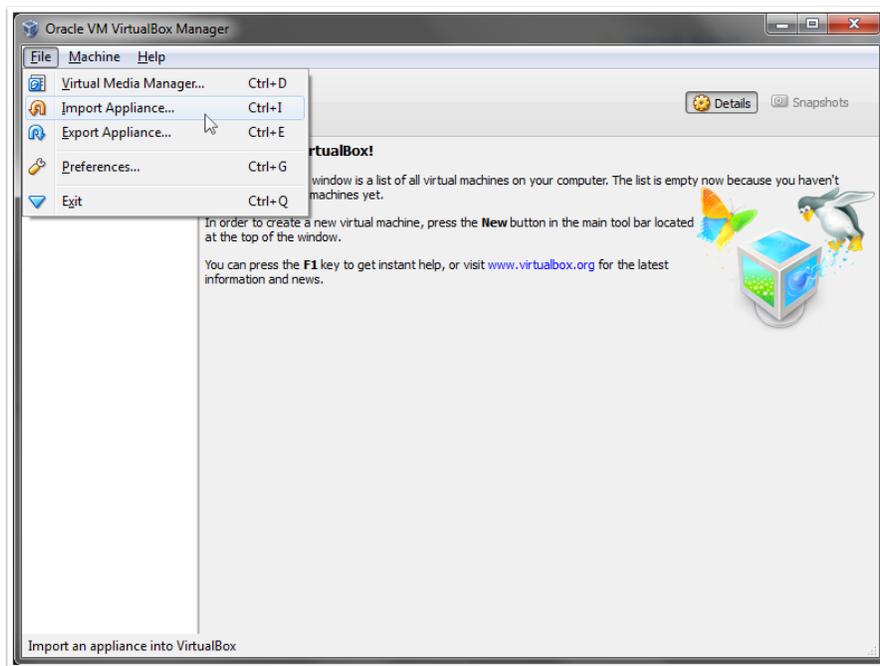


Figure 14 - Click on File->Import Appliance.

When the "Appliance to Import" window appears, click the button to "Choose a virtual appliance to import", and navigate to where the appliance was downloaded (typically in the **Downloads** folder). When you have selected the .ova file, click open. Then back in the "Appliance to import", click **Next**.

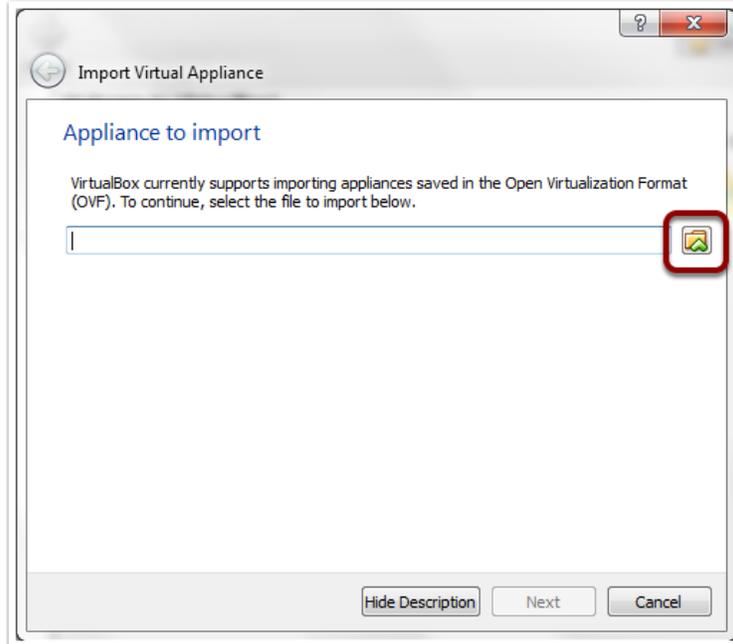


Figure 15 - Click on the folder icon to browse for the downloaded appliance file.

A window with "**Appliance Settings**" should appear with information about the appliance. Note that the values that are displayed on your windows might differ slightly from the values shown in the figure below. Just click **Import**.

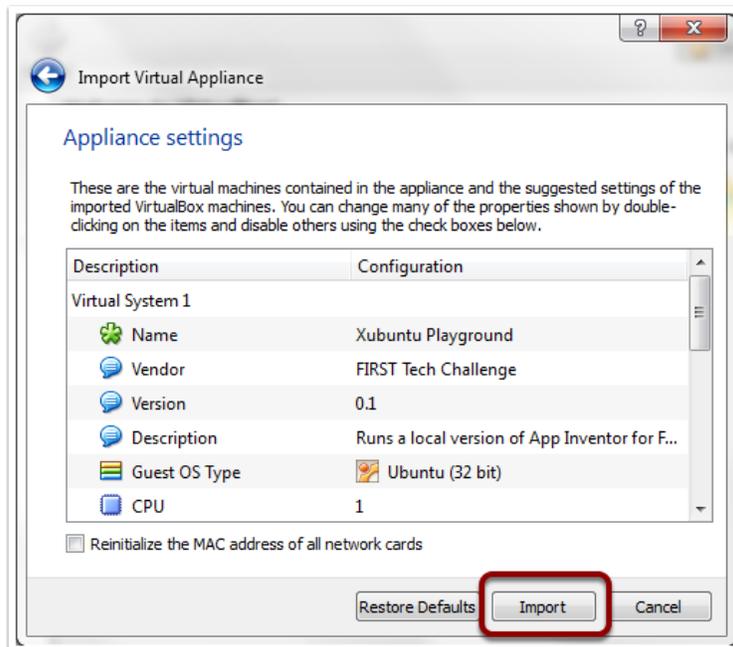


Figure 16 - Click Import to import the Appliance file.

A window will then pop up showing the progress as it imports the appliance. Just wait until it has finished.

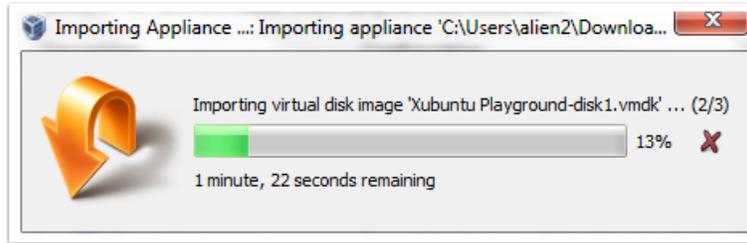


Figure 17 – A Status bar will appear as the appliance is imported.

2.12 Start the Appliance

After the Appliance has imported, it will bring you back to the **VirtualBox Manager**, and the FTC appliance should now show up on the left hand side of the window. To start the appliance for the first time, and any time after that, make sure the **FTC appliance is highlighted**, and then click **Start**. The appliance may take a few minutes to boot, so be patient.

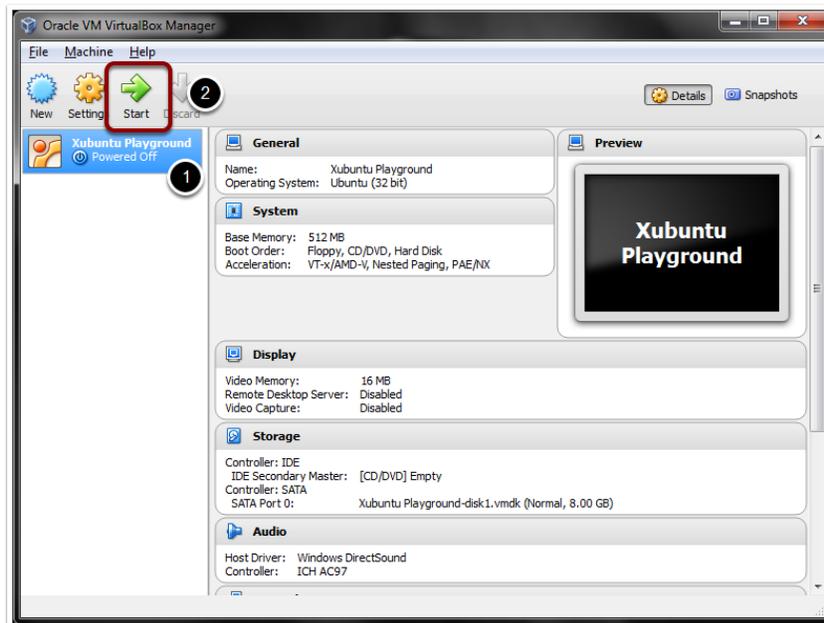


Figure 18 - Select the App Inventor appliance and hit the Start button.

As your appliance is booting up a new window will appear. This window will eventually display the virtual desktop for the App Inventor server. You can use this window to interact with the virtual server.

DRAFT: Contents Subject to Change

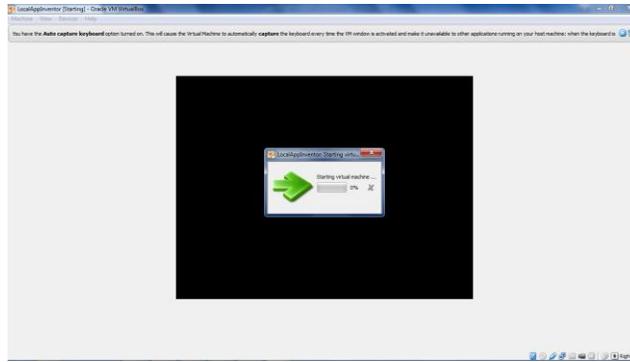


Figure 19 - A new window should appear. This Window will be the desktop interface to your virtual server.

The virtual server’s desktop looks like an ordinary Linux user interface, except it is running within the VirtualBox window. The virtual server’s desktop displays three important pieces of information, the App Inventor server IP address, the App Inventor server status, and the build server status. You will need to know the App Inventor server’s IP address in order to be able to run the application. Also, it is convenient to be able to check on the App Inventor server’s status and the build server’s status. If a server’s status reads “Server:XXX” instead of “Server:ok”, then you might need to wait a few moments until the server has finished loading.

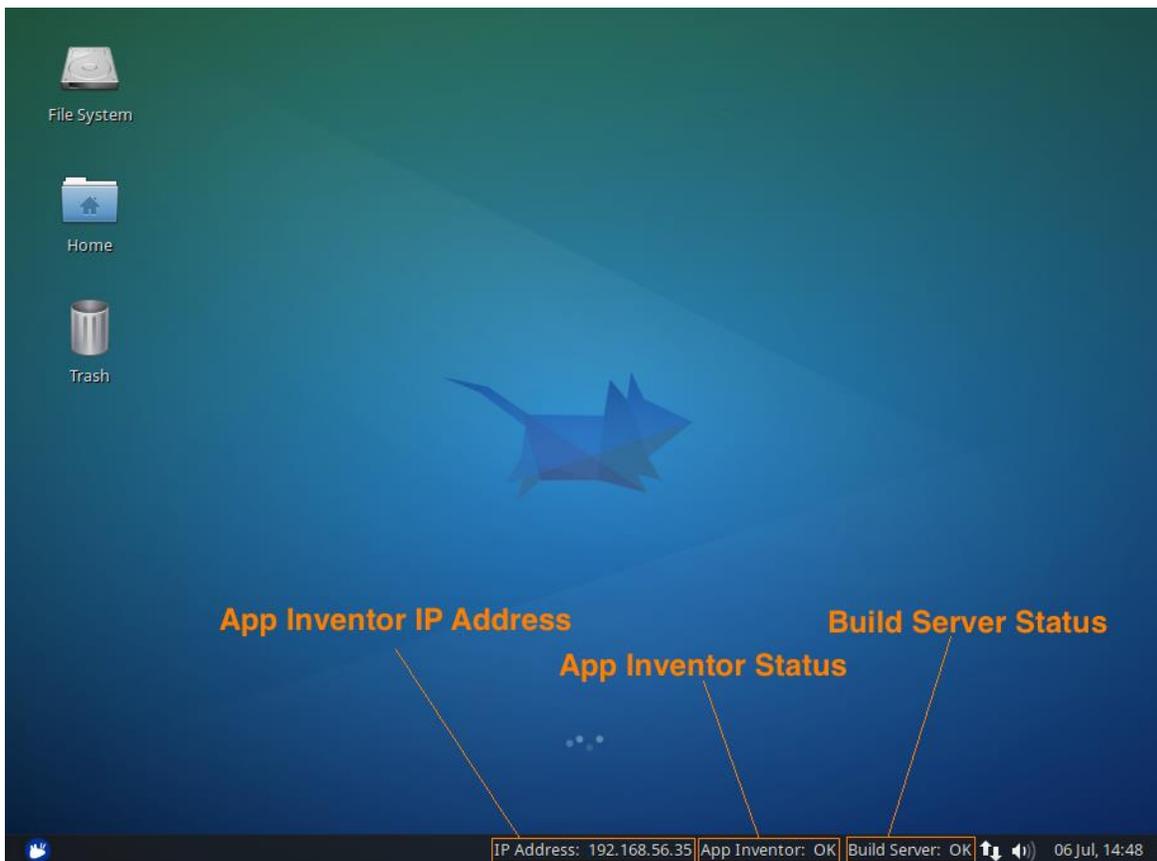


Figure 20 - The virtual desktop displays important info (IP address, App Inventor server status and build server status).

DRAFT: Contents Subject to Change

Note that your virtual server has a password protect user account. If your virtual desktop is inactive for a while, the screen saver lock will appear on the virtual desktop. If you are prompted to unlock the device, use “ftc” as the user password.

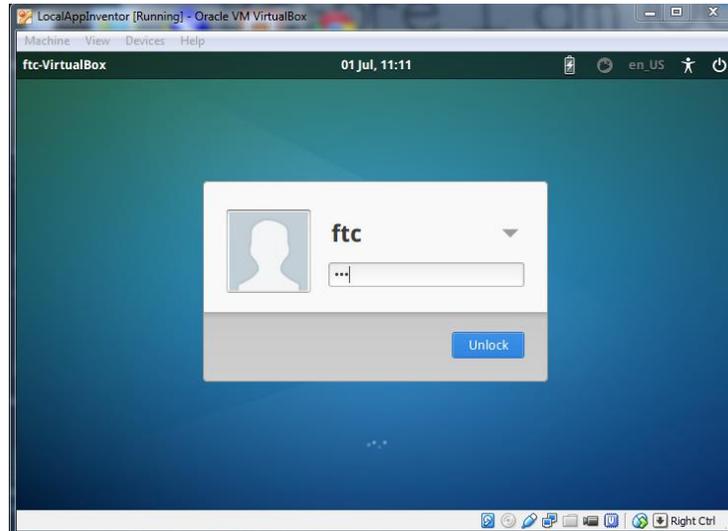


Figure 21 - If prompted use the password "ftc" to unlock the virtual desktop.

2.13 Accessing the App Inventor from your Local Computer

If you have the IP address from the virtual server, and if the App Inventor server status is OK, you are ready to start using App Inventor Locally! To start, launch your Google Chrome Browser. You will need to specify the IP address of your App Inventor server and a port number of 8888 to access the server. In the example in Figure 20 the App Inventor’s server IP address is “192.168.56.35”. Note that this should be the same address for your local App Inventor server. We want to tell our web browser to visit this address and connect to port 8888 at that address.

In your browser window, type "**http://192.168.56.35:8888**", and hit the **Enter** key.

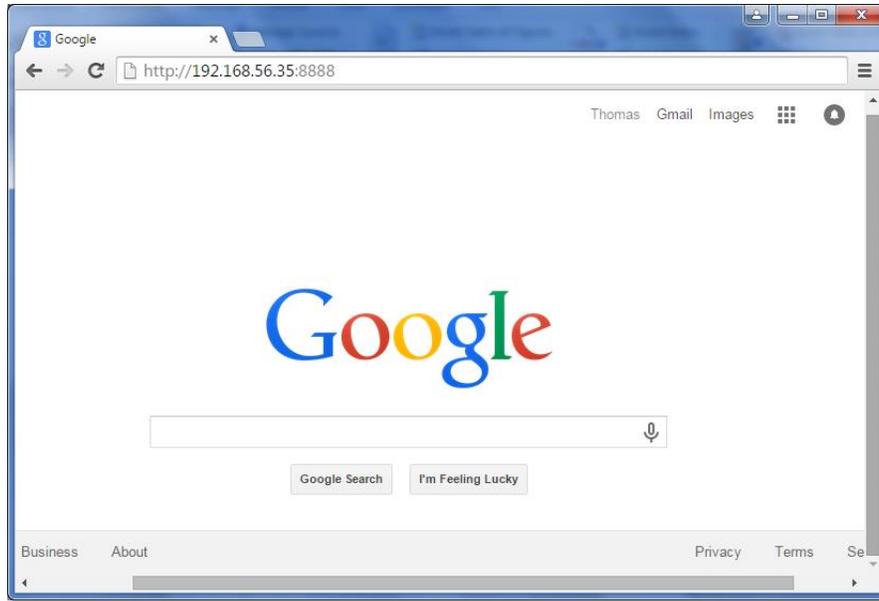


Figure 22 - Type "http://192.168.56.35:8888" into your browser and hit the Enter key.

When you first connect to the app inventor server, it will prompt you to login. You do not have to specify your e-mail. You can accept the default e-mail ("test@example.com") and click on **Log In** to launch the App Inventor. Note that your user profile and list of projects will be associated with whatever e-mail address you use to log in. You can use the default e-mail account ("test@example.com") or login with a different address.

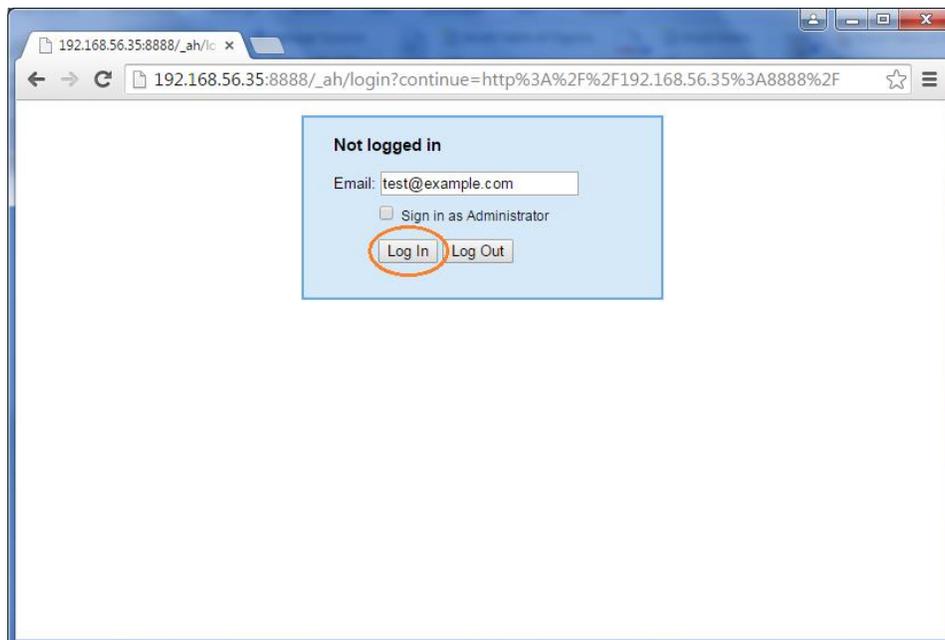


Figure 23 - Hit the Log In button to start using App Inventor locally.

DRAFT: Contents Subject to Change

If a window pops up asking you to accept the **Terms and Conditions**, click **Accept**, and it should bring you to the screen below.

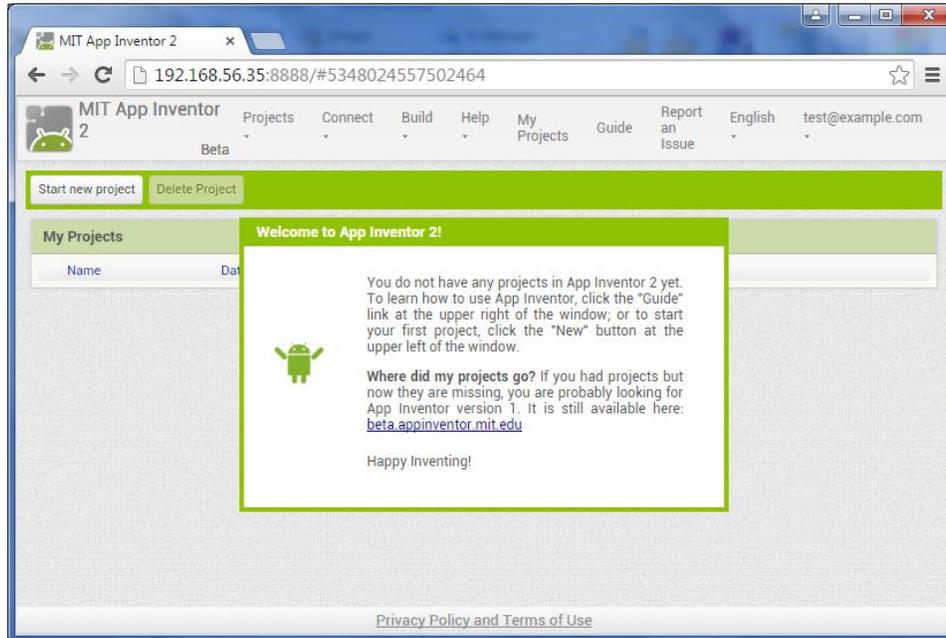


Figure 24 - Congratulations! You did it! You're able to run App Inventor locally!

Congratulations! You did it! You're now able to run App Inventor locally. When you have completed the design of your app, you can build the app and download it as a .apk file onto your local Windows computer's hard drive. You can then use a USB cable to transfer the file to your Android device and to install and run the app on your Android device.

2.14 Shutting Down the App Inventor Appliance

If you no longer need the App Inventor appliance running in the background, you can shut it down from the virtual desktop. If the desktop is locked, use the password "ftc" to unlock it (see Figure 21). To shut down the virtual server, first click on the menu icon in the lower left hand side of the virtual desktop (see Figure 25, label "1") and then press the power button (see Figure 25, label "2") to start the shutdown process.

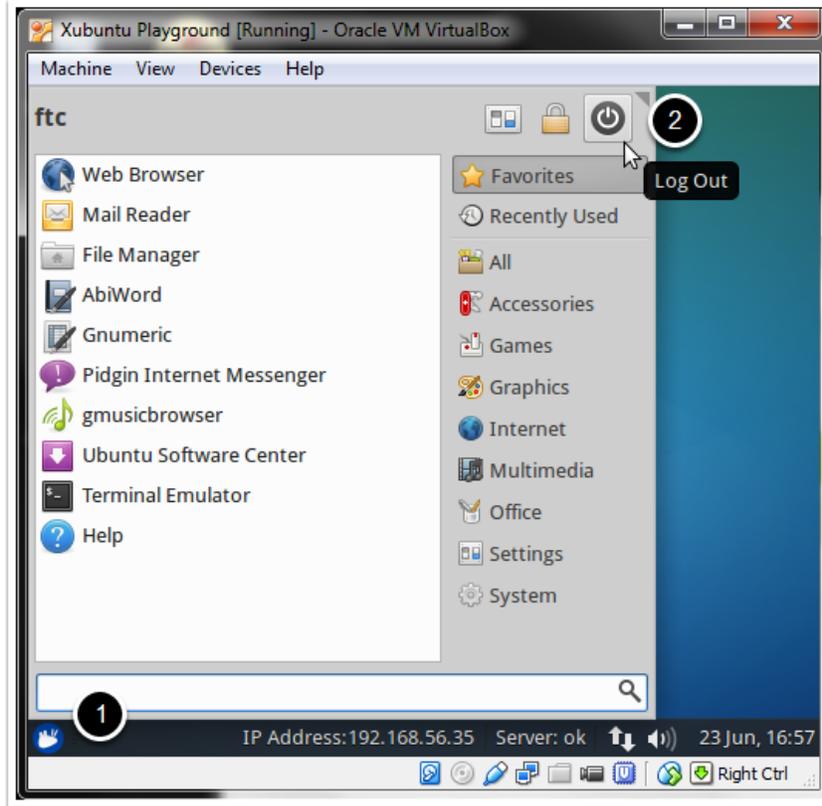


Figure 25 - Press the menu icon (see #1) and then press the power button (see #2) to shut down the virtual server.

A window should appear with a button labeled “**Shut Down**” on it. Press the “**Shut Down**” to turn off your virtual server. If you need to start the appliance again, simply follow the instructions in section 2.12 to restart the virtual server.

3 Let’s Build an App!

3.1 Navigating the Designer Mode Screen

If you are able to see a screen that looks something like the one displayed in Figure 24, then you have successfully configured your local copy of the App Inventor. You are now ready to start making your own apps. Let’s begin by making a very simple app. This will help us learn some of the features of the App Inventor. Note that some of the screen grabs in this manual were made using an Apple Mac computer, but the procedure used on a Windows PC is very similar to the procedure for a Mac.

Let’s begin by creating an App Inventor project. On the welcome screen, press the “Start new project” button located near the upper left hand side of the screen.

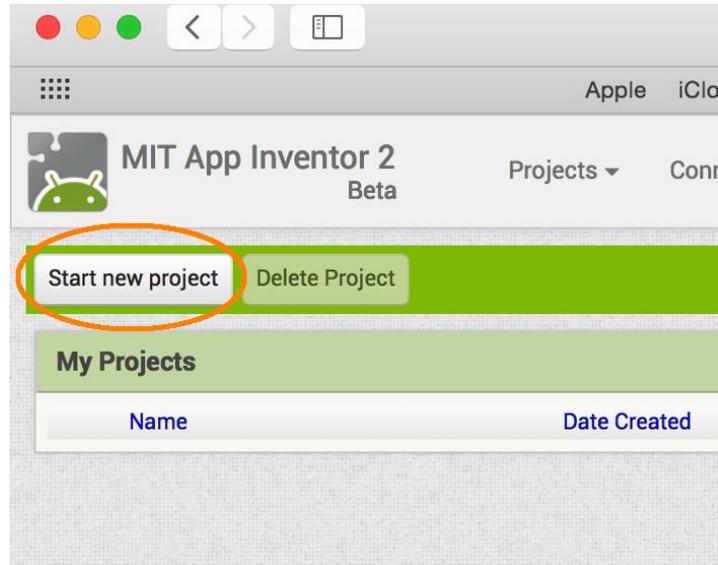


Figure 26 - Click on the "Start new project" button to create your first project.

The App Inventor server should prompt you for a project name. Specify a project name and hit "OK" to create a new project.



Figure 27 - Specify a project name and hit "OK" to create a new project.

Your web browser should open your newly created project in *Designer* mode. Designer mode is the mode of the application where you visually create your Android app using a drag-and-drop visual editor.

DRAFT: Contents Subject to Change

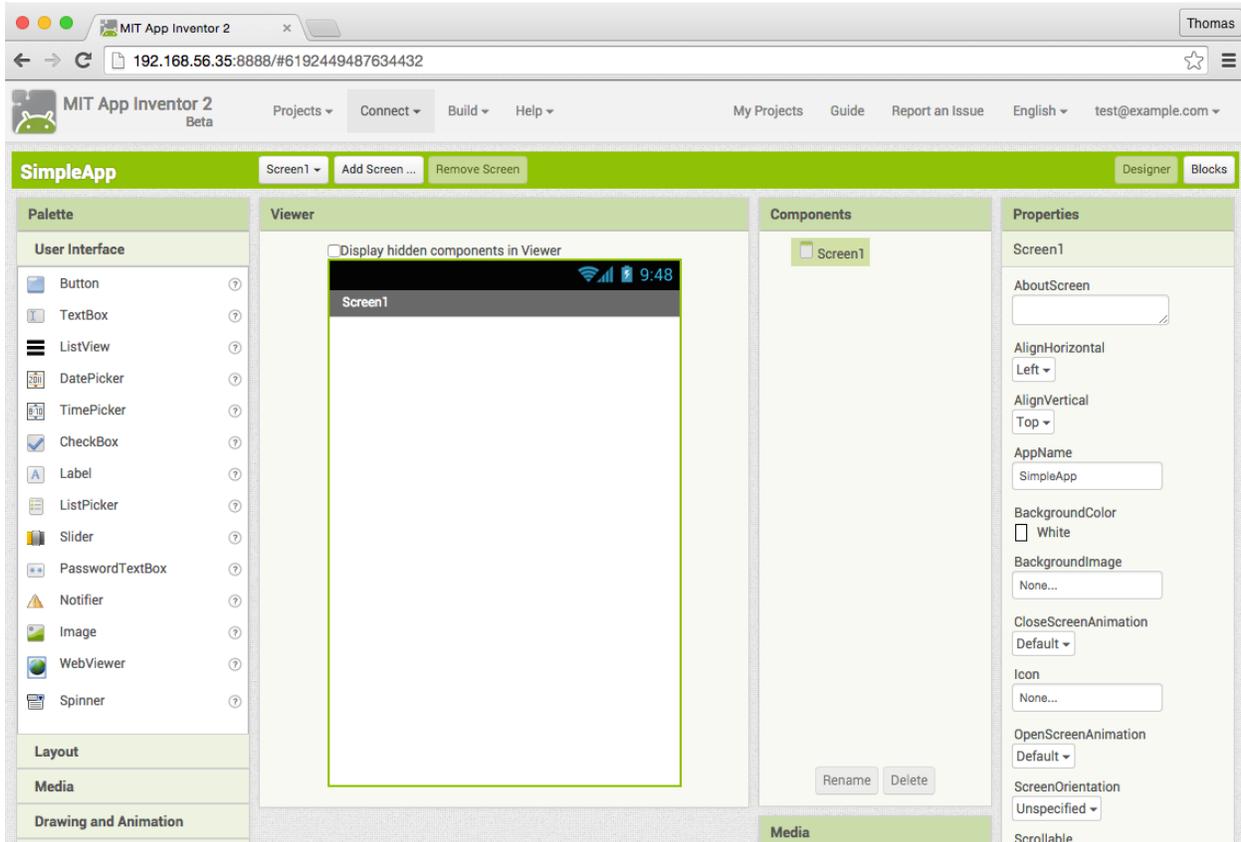


Figure 28 - Your browser should display the project in Designer mode.

Let's take a look at your screen for a moment. There are several App Inventor menu items listed along the top portion of the screen. We will use these menu items to open and close our projects and to build our apps for installation onto our Android device.

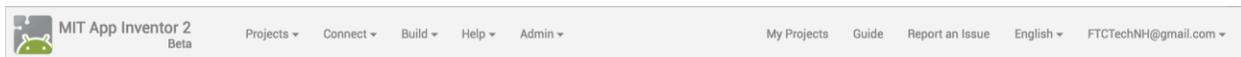


Figure 29 - App Inventor menu items are available along the top of the screen.

On the left hand side of the App Inventor Designer mode screen there is a palette of tools that you can use to add design elements (*widgets*) to your app's screen (also known as an *activity*).

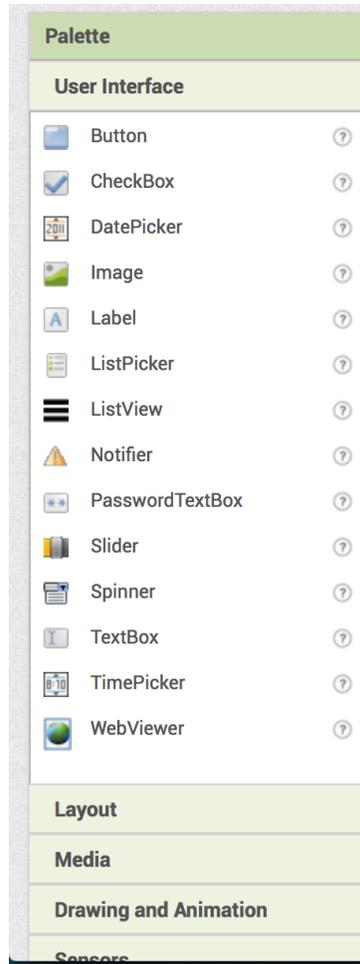


Figure 30 – Several categorized palettes of design elements are available on the left hand side of the screen.

Towards the center of the screen, there is the Viewer pane. In this area of the screen, there is a visual representation of the app's screen or activity. As you design your app, you will drag design elements (such as a button or textbox) from the palette and place them on the screen/activity in the Viewer pane.

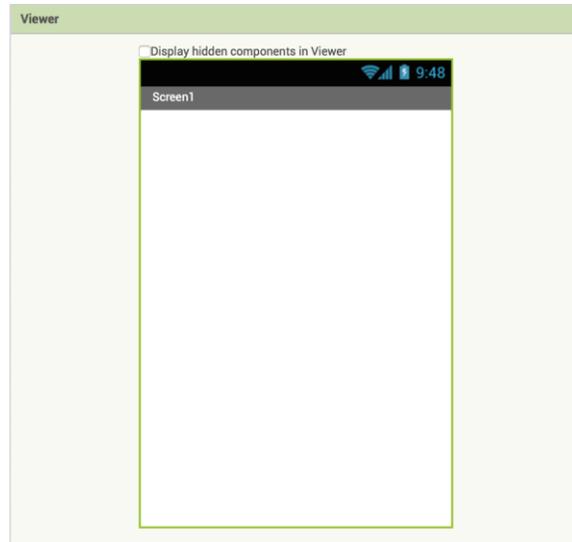


Figure 31 - A virtual screen or activity is visible in the Viewer pane.

The Components pane displays a hierarchical view of the components that make up your app. In the screen shot below, there is only one component listed ("Screen1"). You can select a component and rename it from this pane. You can also select and delete components using this pane.

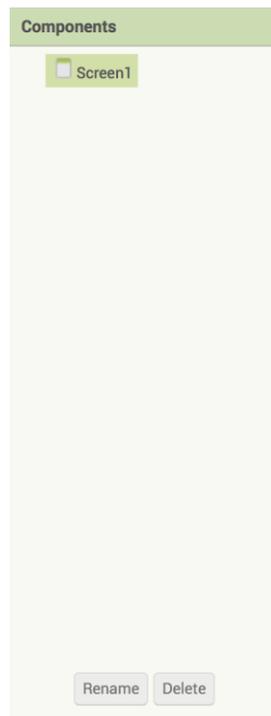


Figure 32 - The components that make up your app are displayed in a hierarchical fashion in the Components pane.

Any component that is selected in the Components pane will have detailed information about the component appear in the Properties pane. This pane shows the properties for the selected component. You can review and edit these properties using this pane.

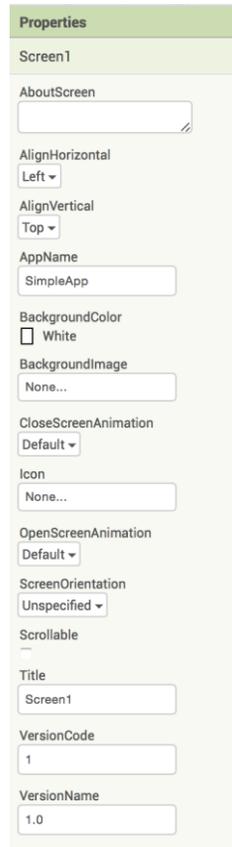


Figure 33 - You can review and edit the properties of a selected component in the Properties pane.

3.2 Creating Our User Interface

The Designer mode of the App Inventor application lets you visually design the user interface for your app. In this example, let's drag a Button design element from the User Interface Palette, and place the button onto our main screen or activity.

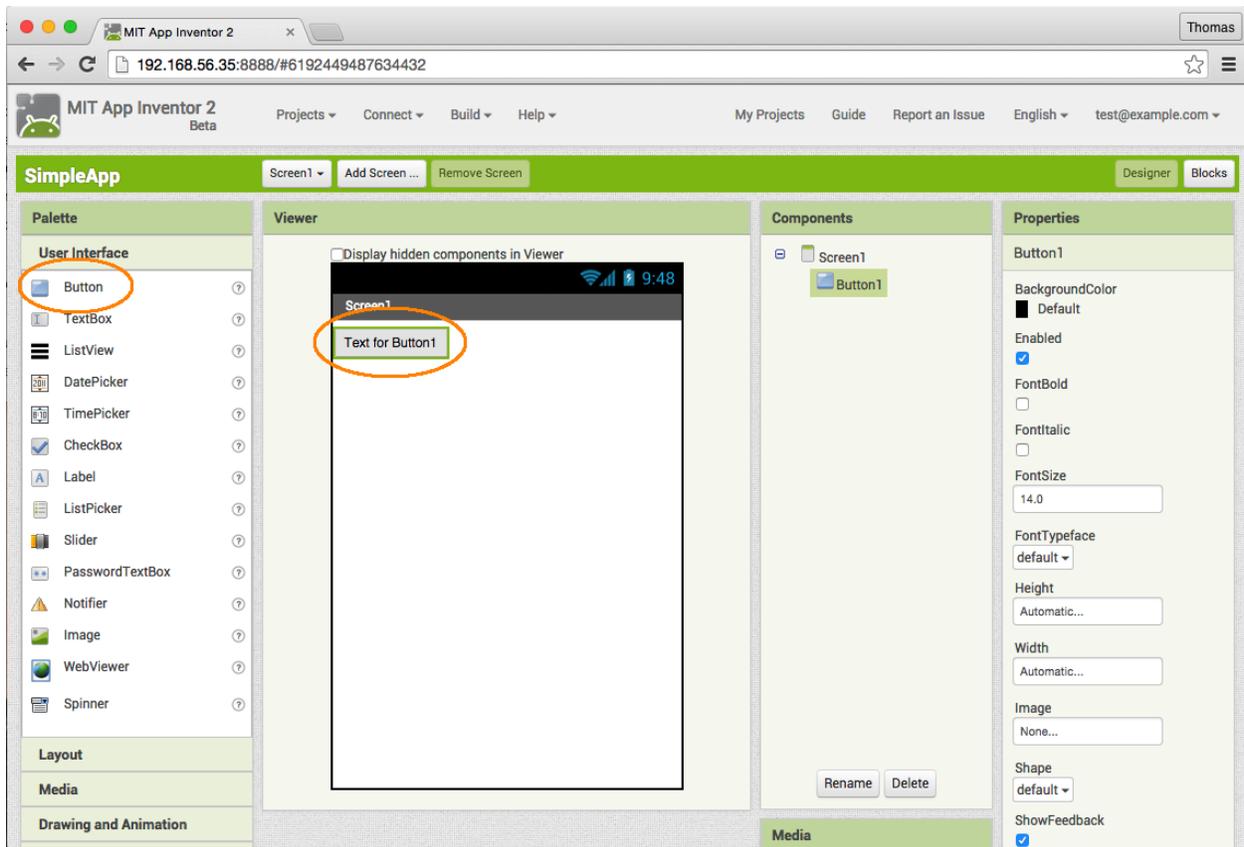


Figure 34 - Use your mouse to select and drag a Button element from Palette to Screen1.

After you have successfully placed a button onto the main screen of the Viewer pane, you should see that there is an element called “Button1” that appears as part of “Screen1” in the Components pane of App Inventor. If you select the “Button1” object in the Components pane, the properties for the “Button1” object will appear in the Properties pane on the right hand side of your web browser.

Let’s rename our button component. In the Components pane, click on the “Button1” object to select it, then hit the “Rename” button.

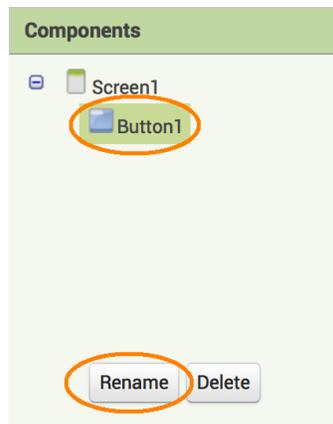


Figure 35 - Select "Button1" and click on the "Rename" button.

DRAFT: Contents Subject to Change

A dialog box should appear prompting you to rename the selected component. Specify a new name of "btnHello" and press the "OK" button to rename your component.

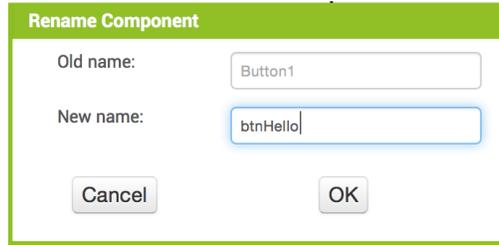


Figure 36 - Rename your component to "btnHello" and press "OK" to accept the change.

After you have renamed the button, let's use the Properties pane to change the text that appears on the face of the button. In the Properties pane, towards the bottom of the pane, there should be a text box labeled "Text" with a default value of "Text for Button1" in the box. Highlight the "Text for Button1" using your mouse and then type in the word "Hello" as the replacement text. You can press the Return key on your keyboard or click an area outside of the text box to commit your change.

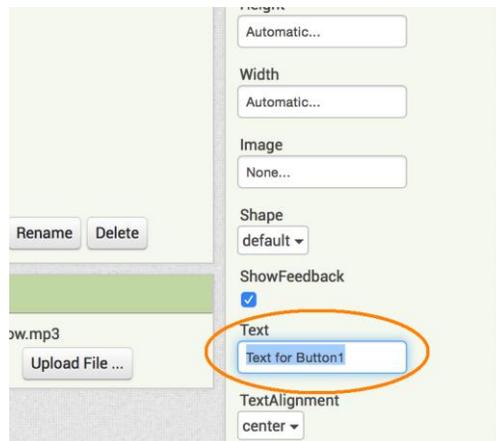


Figure 37 - Highlight the "Text for Button1" text.

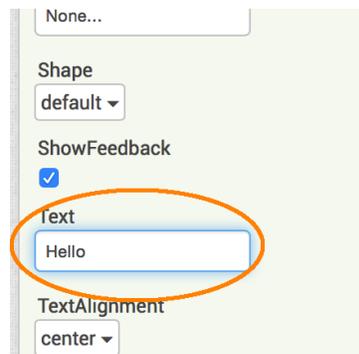


Figure 38 - Replace the text with the word "Hello".

DRAFT: Contents Subject to Change

Once your change has been committed, the button in the Viewer pane should now be labeled with the word “Hello”.

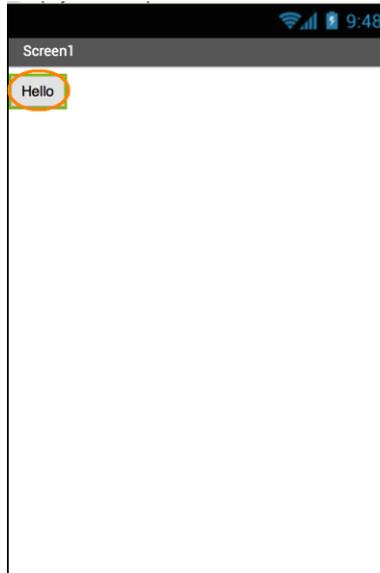


Figure 39 - The button text should now read "Hello" in the Viewer pane.

Now that we have successfully added a button to our user interface, let’s add one more additional design element. We are going to add a “non-visible” element to our main screen or activity. We will drag this element from the User Interface Palette and drop it onto our screen in the Viewer pane. However, this element will not be visible to the user when they run the app. It is a non-visible component that will be used by the app, but will be hidden from view from the user during runtime.

Use your mouse to select and drag a Notifier component from the User Interface Palette to the Viewer.

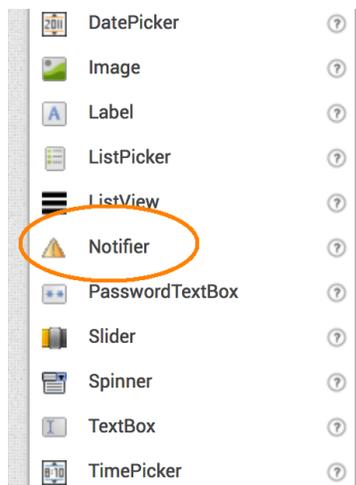


Figure 40 - Drag a Notifier element from the Palette to your app's screen in the Viewer pane.

The Notifier object will appear as “Notifier1” at the bottom of the Viewer pane.

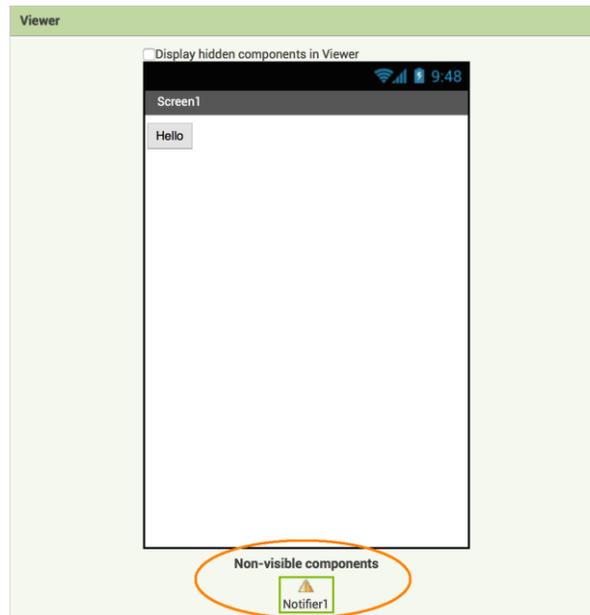


Figure 41 – Notifier1 should appear at the bottom of the Viewer pane as a "Non-visible" component.

If you view Notifier1’s properties in the Properties pane, you’ll notice that it has three properties: BackgroundColor, NotifierLength and TextColor. We will leave the default values for these properties. Note that the NotifierLength determines how long our message will be displayed on the screen. We would like this message to be visible for a long time, so we want to make sure the NotifierLength is set to “Long” in the Properties pane.

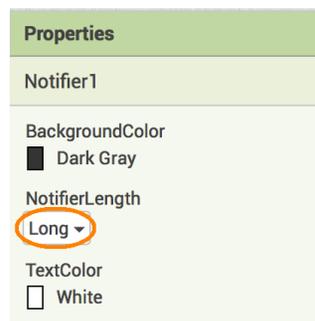


Figure 42 – NotifierLength determines how long our message will be visible to the user.

Now that we’ve placed the Notifier component onto our app activity, we are ready to switch from Designer mode and into Blocks mode.

3.3 Programming in Block Mode

Towards the upper right hand corner of your web browser, there should be a button labeled “Blocks”. Press the “Blocks” button to switch from Designer mode to the Block programming mode.

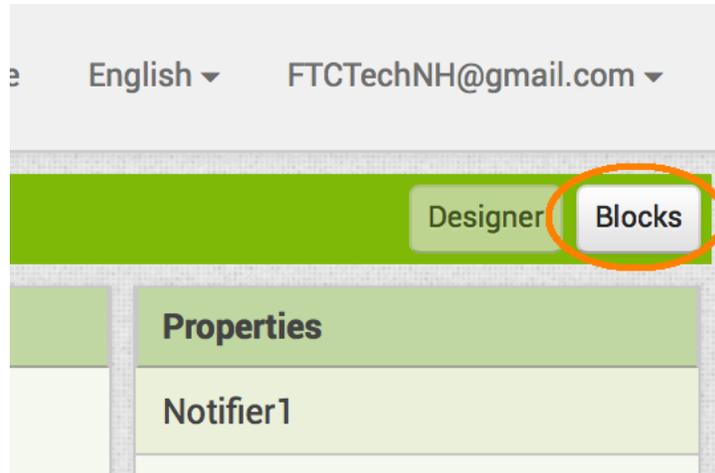


Figure 43 - Press the "Blocks" button to switch to Block programming mode.

We used the Designer interface to design the user interface for our app. Now we need to use the block programming mode to program some logic behind our user interface. The App Inventor uses a Blockly style programming model to let you visually program the logic for your app.

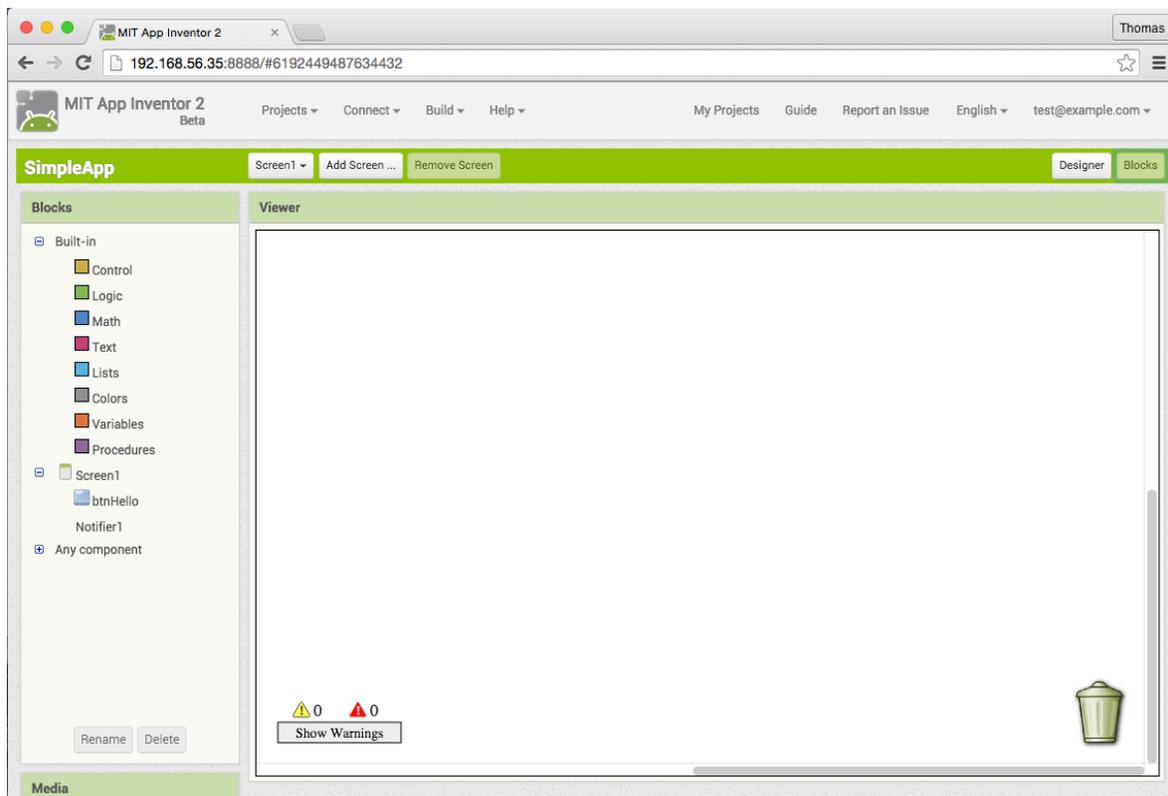


Figure 44 – The “Blocks” mode lets you create the program logic for your app.

Take a look at the programming mode screen. On the left hand side you have a panel or pane that contains categorized program blocks that you can use for your app.

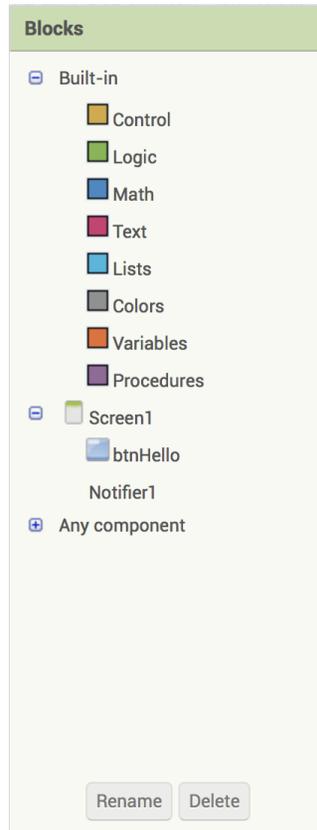


Figure 45 - The left hand pane has a list of categorized built-in and component-specific programming blocks.

You can click on any of the categorized items in the Blocks pane and a screen with available programming blocks will appear. You can drag these blocks onto the Viewer pane to create the programming logic for your app.



Figure 46 - You will assemble your programming blocks in the Viewer pane to create the logic for your app.

Before we begin programming our app, let's think about what we would like our app to do. Our app will be a very simple one. Whenever a user presses or "clicks" onto our Hello button, we would like to

DRAFT: Contents Subject to Change

display a simple message (using the Notifier1 component) to the user. Let's see how we can do this with the App Inventor.

In "Blocks" pane of your programming area, click on the "btnHello" component (which should be visible under the "Screen1" component) to display a list of programming blocks that are associated with that component.

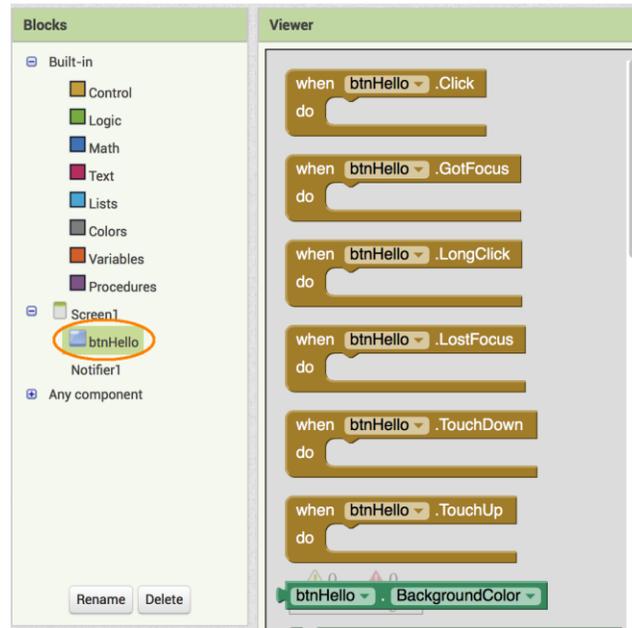


Figure 47 - Click on the "btnHello" component to display the programming blocks associated with this object.

From the list of programming blocks, select the block that has the text "when btnHello.Click" on it and place it to the Viewer screen.



Figure 48 - Place the programming block onto Viewer pane.

This programming block represents the action that occurs when the user clicks on the "btnHello" button. We are going to use this block to tell our app what to *do* when this button is clicked. We want

DRAFT: Contents Subject to Change

to tell our app to use the Notifier component (called “Notifier1” in our app) to display a message (also known as a “toast”) on the screen.

Click on the “Notifier1” object in the Blocks pane to display the programming blocks that are associated with this component.

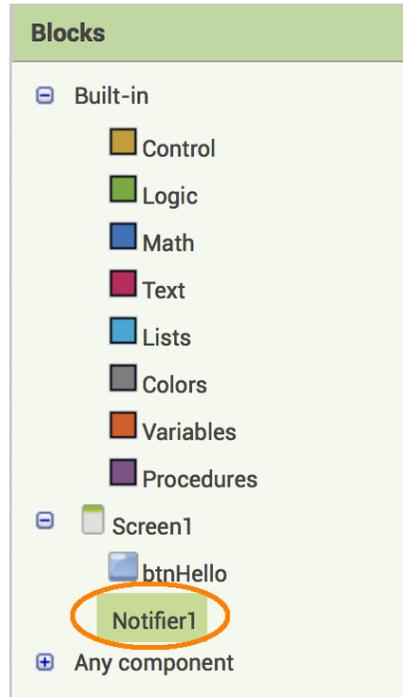


Figure 49 - Click on the "Notifier1" component to display its programming blocks.

When the list of programming blocks appears, scroll down towards the center of the list and look for the block with the text “call Notifier1.ShowAlert” on it. We are going to use this programming block to show an alert message (also known as a toast) on the user screen when the Hello button has been clicked.

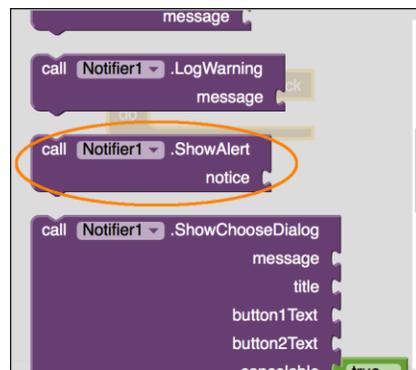


Figure 50 - Scroll down towards the middle of the list and select the "call Notifier1.ShowAlert" programming block.

DRAFT: Contents Subject to Change

Select the “call Notifier1.ShowAlert” programming block and drag it so that you can place this programming block inside the “when btnHello.Click” programming block. Notice that the exterior shape of the “call Notifier1.ShowAlert” programming block matches the shape of the “when btnHello.Click” cutout near the word “do”. Also note that when you place the “call Notifier1.ShowAlert” block near the “do” cutout, the “call Notifier1.ShowAlert” block gets outlined in yellow and it matches the highlighted yellow tab on the “when btnHello.Click” block.

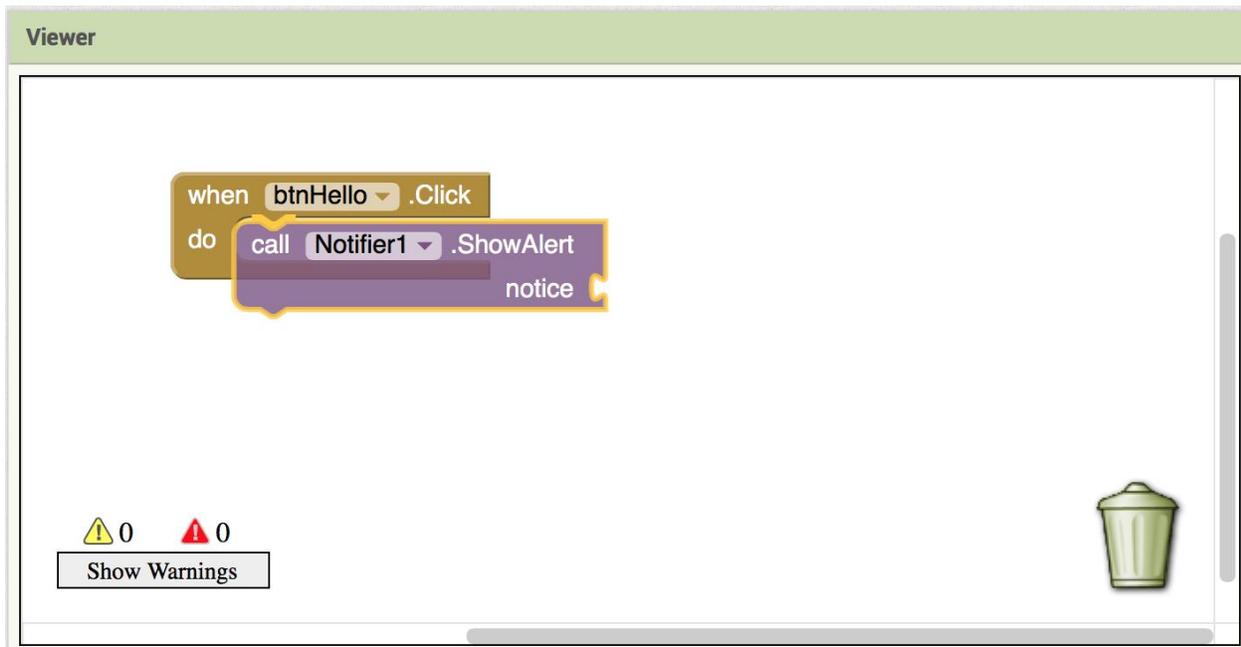


Figure 51 - Place the "call Notifier1.ShowAlert" block into the match space of the other programming block.

Once you have placed the Notifier1 block into the btnHello block, the two should interlock like pieces of a puzzle. Let’s take a look at our programming logic for our app so far. When a user clicks on the Hello button, the app will display an alert message to the user. Now we have to specify what message or “notice” we want to display.

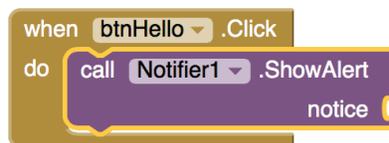


Figure 52 - When a user clicks on the Hello button, the app will display an alert message to the user.

DRAFT: Contents Subject to Change

The last thing we need to do is to specify the text string that will be displayed in the alert message. From the Blocks plane, select the “Text” item from the list to display the text-related programming blocks.

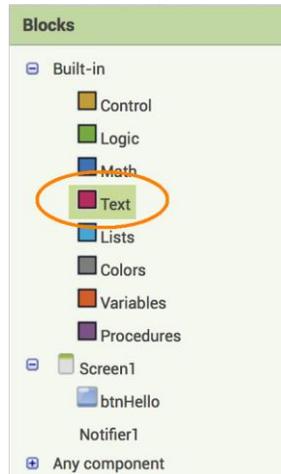


Figure 53 - Select the "Text" item to display text-related programming blocks.

Find the programming block labeled with the quotation marks (see) and drag this block and interlock it to the “notice” side of the “call Notifier1.ShowAlert” programming block.

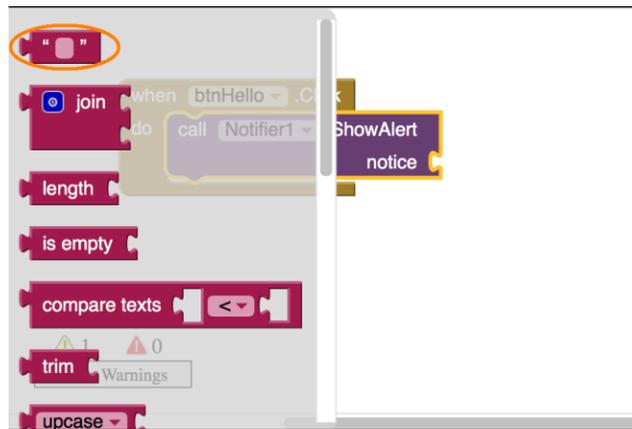


Figure 54 – Select the double quotation mark block and connect it to the Notifier1.ShowAlert block.

Once the blocks interlock, click on the text box in between the quotation marks and type in the phrase “Hello World!”.



Figure 55 –Type in the phrase “Hello World!” in between the quotation marks.

Let's review the programming logic... when a user clicks on the btnHello component, the app will call the ShowAlert method of the Notifier1 component to display a text string with the phrase "Hello World!" to the user.

3.4 Building Your App

Now that we have designed our interface and have created some logic behind the interface, we are just about ready to build our app and install it onto our phones. When we say that we want to *build* our app, we mean that we want to convert the design elements and programming logic that we created using the App Inventor into a package that can be loaded onto and executed by an Android device. The App Inventor will pass information about our design and logic to a *build server*. The build server will convert this information into a format that an Android device can understand and store all of the information needed to run the app in a file that ends with a ".apk" suffix. You can then transfer this .apk file to your Android device and then install and run the app on your device.

Building your app is a very simple process. You simply select a menu item from the App Inventor console and the App Inventor server does all of the work. You will then transfer the .apk file to your Android device after the app has been created.

3.4.1 A Note Regarding App Inventor Local and Using a QR Code

Normally, if you are using the server version of the App Inventor, you can tell the App Inventor to build your .apk file and then generate a QR code that your Android device can scan in to locate and download the .apk from the server. Unfortunately, for the local version of App Inventor, this feature is unavailable. Because the VirtualBox appliance is accessed through a host-only connection (i.e., a network connection that is only available to your PC or Mac running VirtualBox) your Android device will not be able to access the .apk file stored on the App Inventor server. Instead, you will have to use a USB cable to transfer the app to your phone.

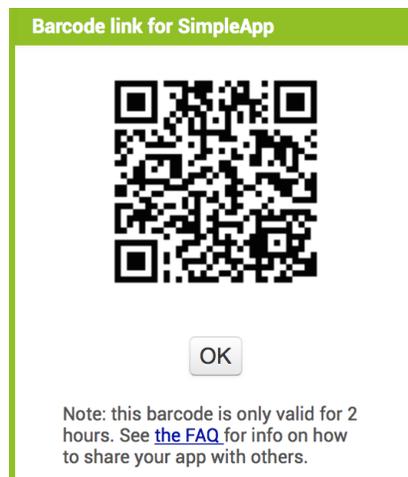


Figure 56 - An Android device will not be able to access the .apk file using a QR code for the local App Inventor.

3.4.2 Using a USB Cable to Install Your App

You will need to install the app onto your Android device using a USB cable connected between your Android device and your laptop. Towards the top of your browser screen look for the “Build” menu item and select it with your mouse. From the pop up menu that appears, select the “App (save .apk to my computer)” option. The App Inventor server will initiate the build process and it will download the .apk file to your computer.

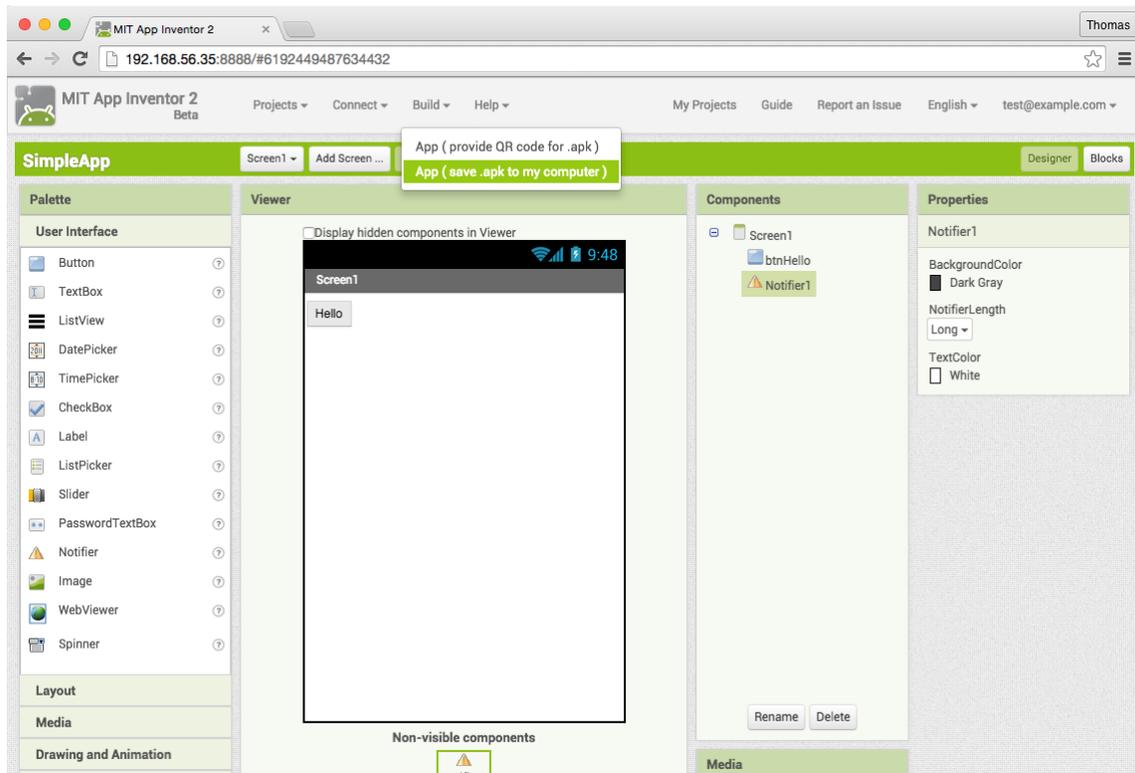


Figure 57 – Select Build-> App (save .apk to my computer)

A progress bar should be displayed on the screen as the build server builds the app. It may take a few minutes to complete the build.

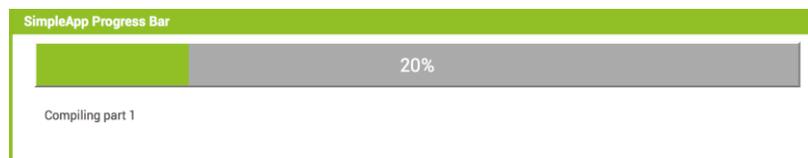


Figure 58 - A progress bar will be displayed as the app is being built.

Once the .apk has been built, it should automatically be downloaded to your computer.

DRAFT: Contents Subject to Change

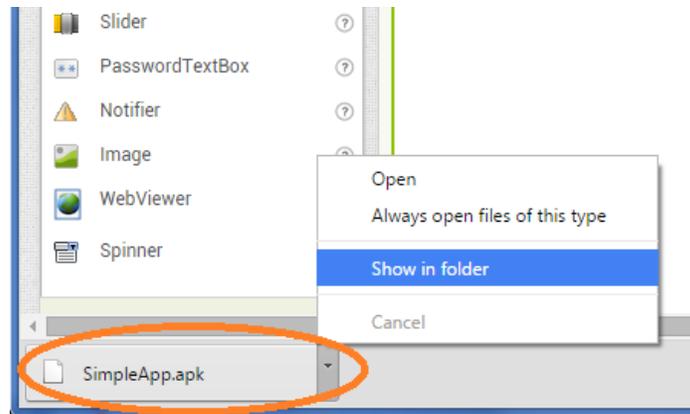


Figure 59 - The .apk should automatically be downloaded to your computer.

Use your laptop's file explorer to locate the app on your laptop.

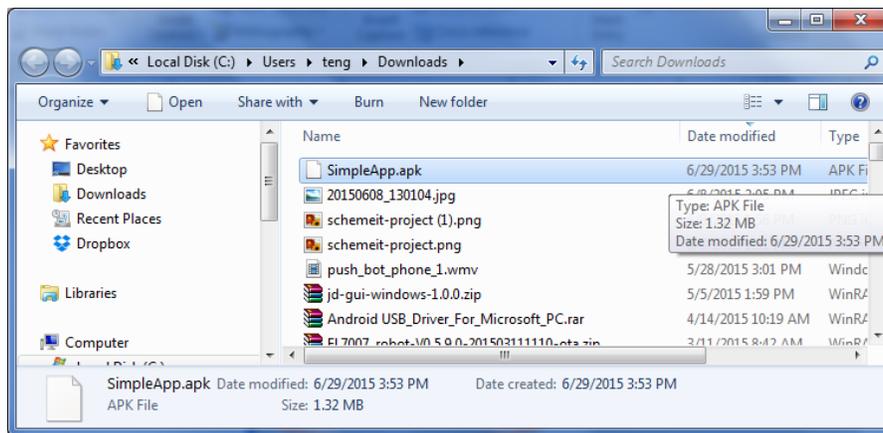


Figure 60 - Use your computer's file browser to locate the .apk file.

Now that your .apk file has been built and downloaded, you will need to transfer it to your Android device. If you are using a ZTE Speed phone, start by connecting the computer to the phone using a USB cable.



Figure 61 - Connect the ZTE Speed with a USB cable.

DRAFT: Contents Subject to Change

Once you have connected the ZTE Speed to your computer, you want to check to make sure your phone is in Media Device mode. To check if your phone is in Media Device mode, use your finger to slide down from the top of the home screen to access the USB options menu.

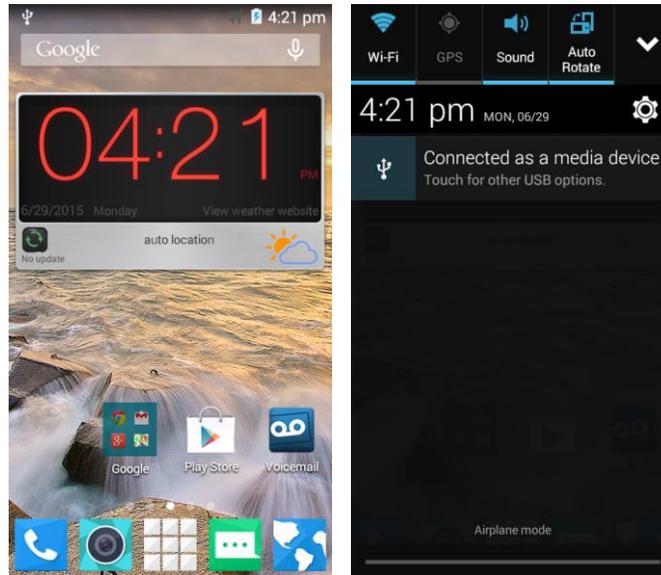


Figure 62 - Touch the top of the screen and slide down to access the USB options menu.

Touch the “Touch for other USB options” item launch the USB options menu. In the USB options menu, you want to make sure your phone is in “Media device (MTP)” mode. This means that your phone will look like a USB media device (such as a USB thumb drive) to your computer.

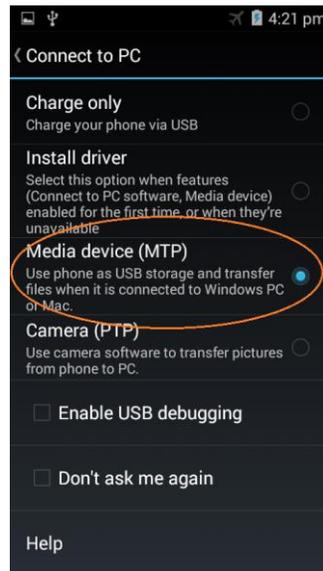


Figure 63 - Make sure your phone is in Media device (MTP) mode.

DRAFT: Contents Subject to Change

Your phone should look like a media device in your computer's file browser. Copy and paste the .apk file your PC to the **Download** directory of your Android device.

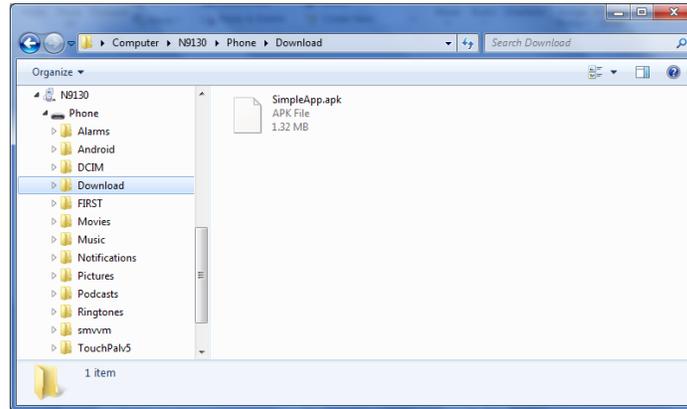


Figure 64 - Copy and paste the .apk file to the Download directory of your ZTE Speed phone.

Once you have the .apk file copied to your phone, you want to use the phone's **File Manager** to find the file on your phone and to install it. Go to the Apps listing on your ZTE Speed phone and look for the app named "File Manager".

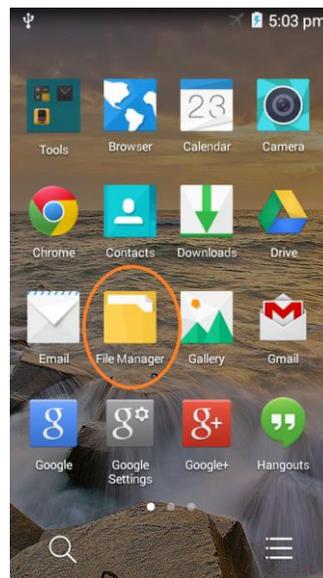


Figure 65 - Launch the File Manager so you can browse for the .apk file.

Use the File Manager app to locate the .apk file in the Download directory of your ZTE Speed phone. You need to click on the **PHONE** tab at the top of the File Browser app and then look for the **Download** folder. Click on the **Download** folder to open it. You should see the **SimpleApp.apk** file within the folder. Click on the .apk file to install it.

DRAFT: Contents Subject to Change

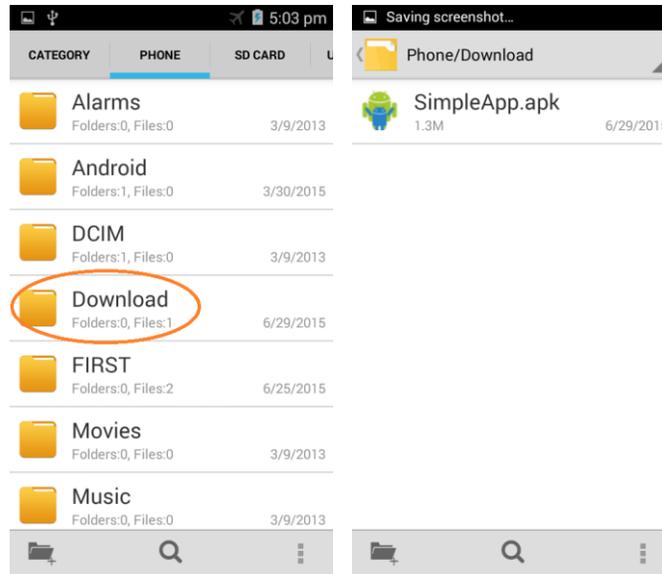


Figure 66 - Use the File Manager to locate the .apk. Click on the .apk file to install it.

Depending on your security settings, your ZTE Speed phone might display a security warning, indicating that the app installation was blocked since the app was not downloaded through the Google Play store. Click on the “Settings” button to display the Security settings. Make sure the “Unknown sources” option is selected for your phone.

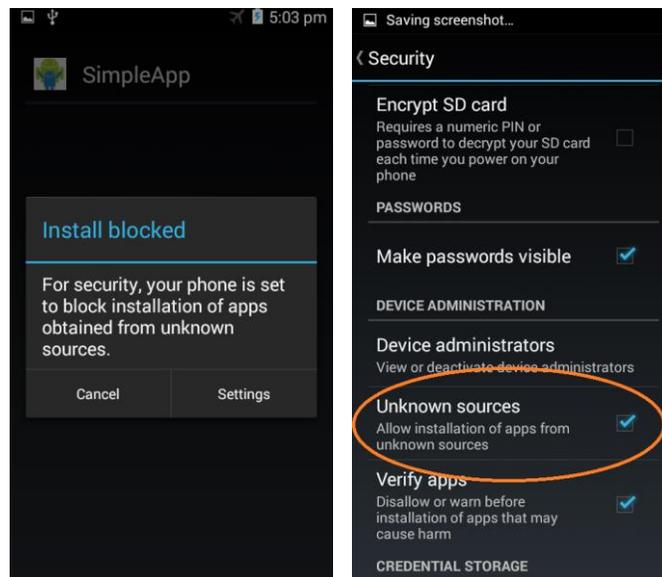


Figure 67 - Go to the Security Settings menu and make sure the "Unknown sources" option is checked.

Once you have verified that your security settings allow for the installation of apps from “Unknown sources” use your back arrow to return to the install screen. Android should prompt you to make sure you would like to install the app onto your phone. Hit the “Install” button to start the installation.

DRAFT: Contents Subject to Change

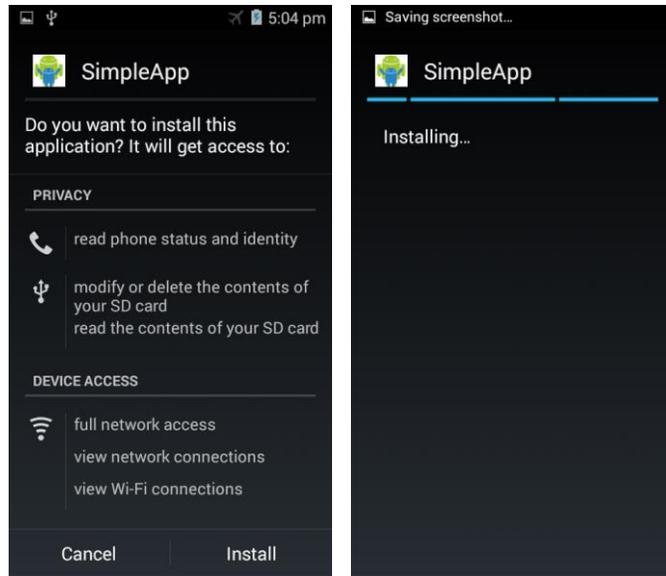


Figure 68 - Click on the "Install" button to start the installation.

Click on the "Open" button to launch the app, once the installation is complete.

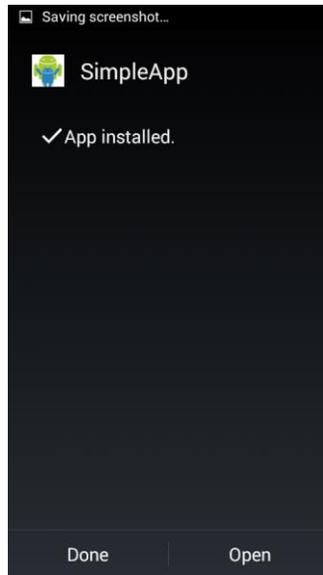


Figure 69 - Click on the Open button to launch the app.

Once the app is launched, press the "Hello" button to display the notification.



Hello World!

Figure 70 – Press the “Hello” button to display the notification.

Congratulations! You just built an app and transferred it to your device using a USB cable!