

The FAIMS Mobile User to Developer Guide: How to Build and Deploy Field Data Collection Systems for Arbitrary Methodologies

Version 1.0.20171013

Dr Brian Ballsun-Stanton

Russell Alleen-Willems

Adam DeCamp

Funded by the NeCTAR V005 Documentation grant for RT043 (FAIMS Project).

Typeset in ConT_EXt.

Code for this project and the latest version of this document can be found at

<http://github.com/FAIMS/UserToDev>.

This document is licensed under CC-BY-SA International 4.0, Copyright 2017.

1	Making Your Own Modules	3
1.1	Deploying the Virtual Machine	3
1.1.1	Hardware Assumptions	3
1.1.2	Before You Start	4
1.1.3	Installing the FAIMS Server and Virtual Machine (VM)	4
1.1.4	Enable VM extensions in your computer BIOS for better VirtualBox Performance (VT Hypervisor)	4
1.1.5	Installing VirtualBox	5
1.1.6	Getting the FAIMS Server Up to Date	16
1.2	Installing the FAIMS module “Auto-gen”	17
1.2.1	Installing the prerequisites for the Autogen	18
1.2.2	Installing the FAIMS Tools	19
1.2.3		20
1.2.4	Installing a Text Editor for Editing Necessary Files	20
1.2.5		21
1.2.6	Get Access to your local files via VirtualBox	21
1.2.7	Installing the Android SDK	21
1.2.8	Installing the FAIMS Debug APK on your Android Device	22
1.2.9	Making your Android Device Communicate with your Computer with USB Debugging	25
1.2.10	Test Your Knowledge	26
1.3	How to Code Modules	26

1 Making Your Own Modules

1.1 Deploying the Virtual Machine

Before you can code any new modules, you'll need to download some files and free programs. Some of those programs you'll need to configure to make coding new modules easier and more efficient. At the end of this section, you will be able to:

- Run the FAIMS Server, where your team members will download your module from and upload data to, on a virtual machine.
- Examine all the necessary files of a FAIMS module with a text editor.

1.1.1 Hardware Assumptions

Every computer is different, and unless you're using exactly the machine we used when writing these instructions (an HP laptop with a Core i7 processor and 8GB of RAM, running Windows 7) you probably won't be able to follow each step exactly as it's written. Many of the differences will be very minor; an option might have a slightly different name or be located somewhere else on your computer, for example. When you find an option isn't present as we describe it, take a deep breath. Click around or use your computer's "search" feature to see if you can find the option or setting somewhere else. If necessary, web search the terms we use in our instructions along with the system you're using in order to find equivalents. A good rule of thumb for general computer configuration is: whatever you can't figure out right now, someone else has had the same problem and a third person has fixed it for them.

As a last resort, we offer technical support services. See the appendix for further information.

1.1.2 Before You Start

Before starting, you'll need to make sure you have enough room on your hard drive. We recommend about 25GB minimum for the server installation, as well as enough RAM to hold both your current operating system, the emulated machine, and any open programs in working memory. If you don't have that much space, delete files (especially large media files, like unnecessary videos) or uninstall programs until you're ready to begin. No good will come of trying to follow these next steps without enough room to work.

1.1.3 Installing the FAIMS Server and Virtual Machine (VM)

For various reasons, the FAIMS server is designed to run on a machine running an operating system called Ubuntu Linux 14.04 (as opposed to, say, Windows or OSX). You probably don't use Ubuntu, and you probably wouldn't like to use it all of the time. That's why our first task here is to set up a "virtual machine" on the computer you do use. Virtual machines let you emulate entirely different kinds of computer system without making any significant changes to your own—in this case, an Ubuntu Linux 14.04 machine. It's like you're installing an application that simulates an entirely different computer whenever you need it.

1.1.4 Enable VM extensions in your computer BIOS for better VirtualBox Performance (VT Hypervisor)

You'll have a much easier time running your virtual machine, as well as Android emulators you'll need later, if you first enable a feature specifically designed for this purpose called "VT Hypervisor."

To enable VT Hypervisor, enter your computer's BIOS or UEFI menu while booting your computer. This process differs a lot from computer

to computer, but you should be able to find instructions here: <http://www.howtogeek.com/213795/how-to-enable-intel-vt-x-in-your-computers-bios-or-uefi-firmware/>

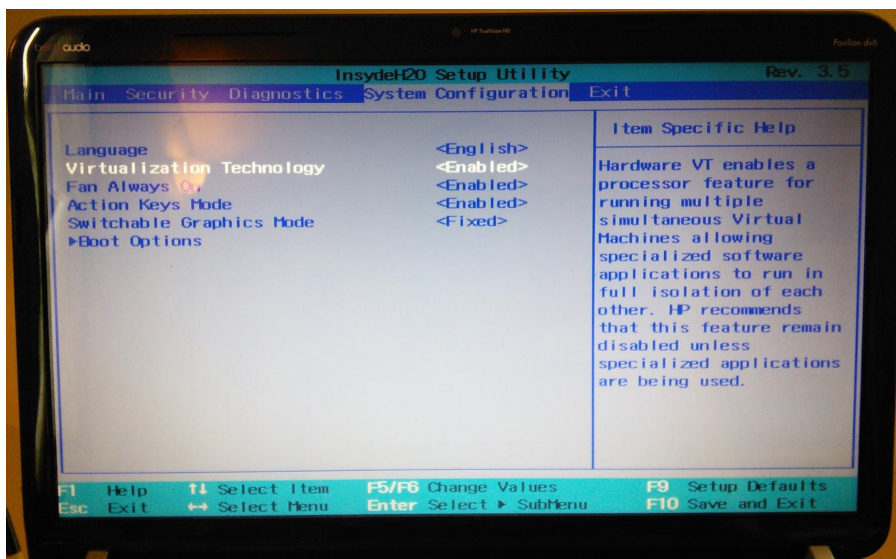


Figure 1.1 A rather literal picture of a computer’s “BIOS” setup screen, showing the enabling of “VT-X extensions”

We found the VT setting under the “Virtualization Technology” option in the System Configuration menu. Again, unless you’re using an HP dv6qt Laptop, yours will probably be somewhere else. This process can feel a little intimidating if you’re not used to messing with your computer on this basic a level, but relax: this step is perfect safe.

1.1.5 Installing VirtualBox

Now you can download and install the VirtualBox client from: <https://www.virtualbox.org/wiki/Downloads>. This is what we're going to use to create a virtual machine that can run our Ubuntu Linux-based FAIMS server.

You probably won't have installed VirtualBox before, but on the off chance you have (say, if this isn't your first time trying this step) make sure you uninstall the old version **completely** before trying to install it again.

During installation, you may receive a couple Windows Security questions about Oracle drivers. Windows is naturally suspicious of this kind of installation, but nothing you're putting on your computer right now is dangerous. You can ignore each prompt individually or skip them all at once by selecting "always trust Oracle" in the popup window (Figure 1.2).

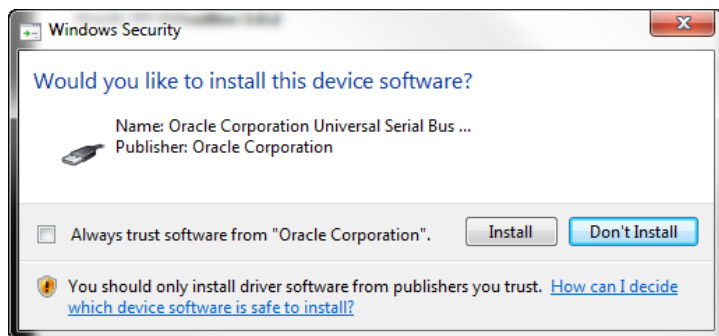


Figure 1.2 We are installing network device drivers here, so that your tablet can talk to this “Virtual Machine”

Once VirtualBox is installed, you can start the program. The default screen should look similar to Figure 1.3

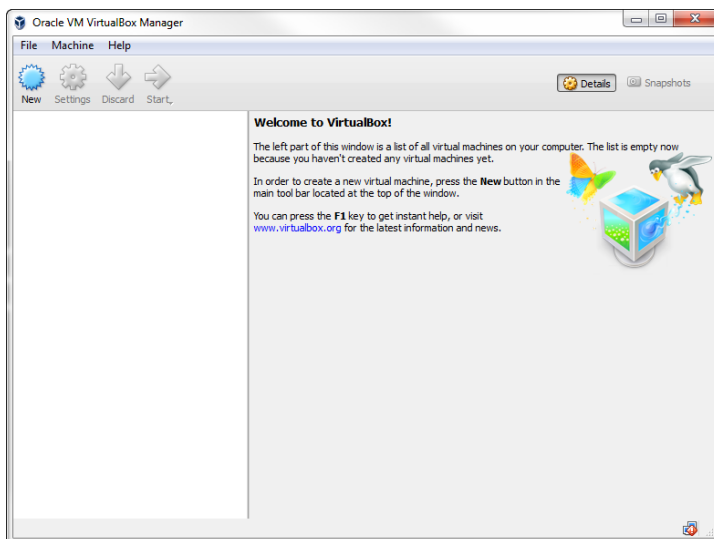


Figure 1.3 VirtualBox without any virtual... boxes. A box inside your computer which can hold boxes¹, but not cats.

If you need to take a break, this is a good stopping point²!

If you're ready to move on, download the FAIMS Server image (2.5GB) from: <https://cloudstor.aarnet.edu.au/plus/index.php/s/0ToMafgHrpsvwTG>

The server image is packaged as an ".ova" file, which is a type of file that can be opened by VirtualBox. This file is the complete package: it represents an Ubuntu Linux 14.04 computer system that has the FAIMS server set to automatically configure and run. This image is the same one that our *FAIMS-In-A-Box kits* use, which will make supporting your FAIMS installation much easier if you run into any problems.

Before using the image, you'll need to unzip the compressed zip file using your favorite archival program (e.g. *7-ZIP*, *ICEOWs*).

¹ We call a computer a box because the old fashioned tower that older or powerful computers come with is usually referred to as the Box by tech types

² Time for coffee! It's *always* time for coffee.

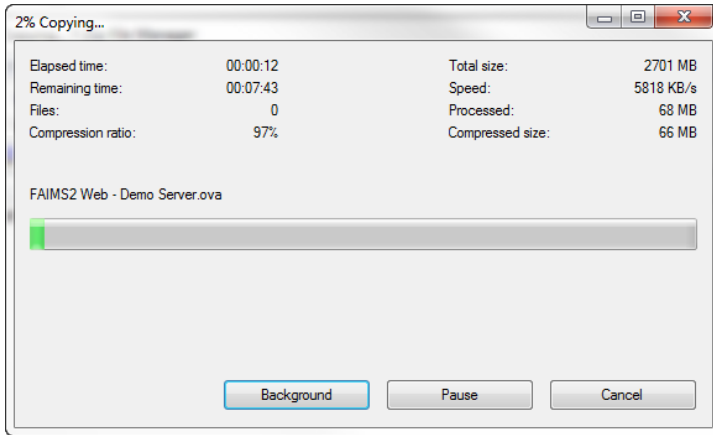


Figure 1.4

To set up the emulated machine, we'll use the preconfigured machine image we downloaded in the last step.

In VirtualBox, select the "File->Import Appliance" option.

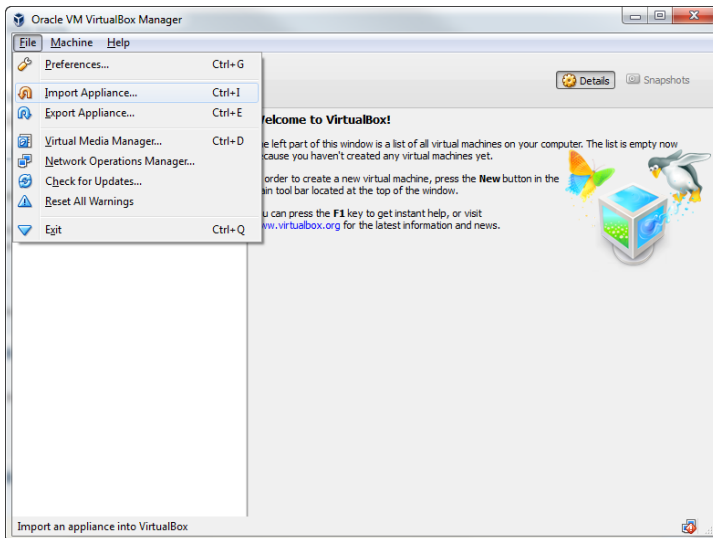


Figure 1.5

Using the file browser that appears, navigate to the directory where you downloaded the FAIMS server image and select to open it.

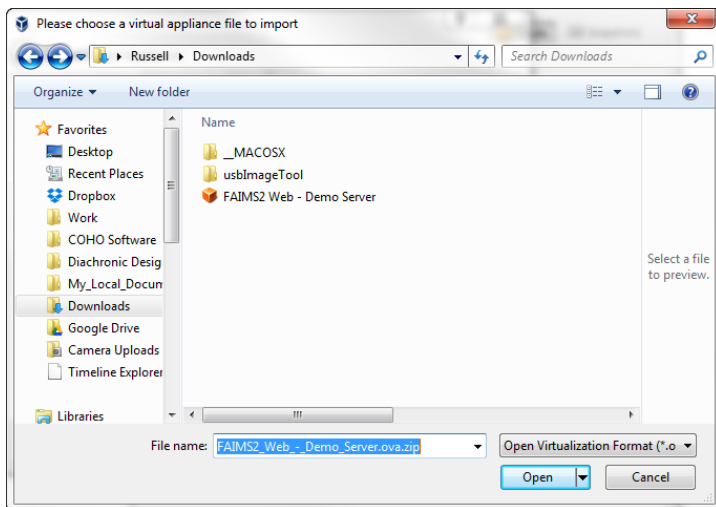


Figure 1.6

The pre-configured settings will appear. Select the “Import” option.

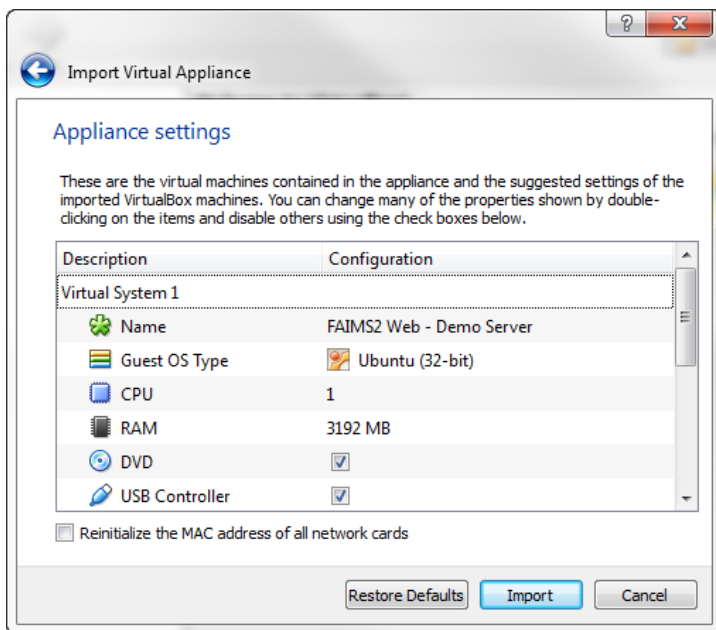


Figure 1.7

VirtualBox will now set to work importing the image and setting everything up. Depending on your system, this may take a few minutes.

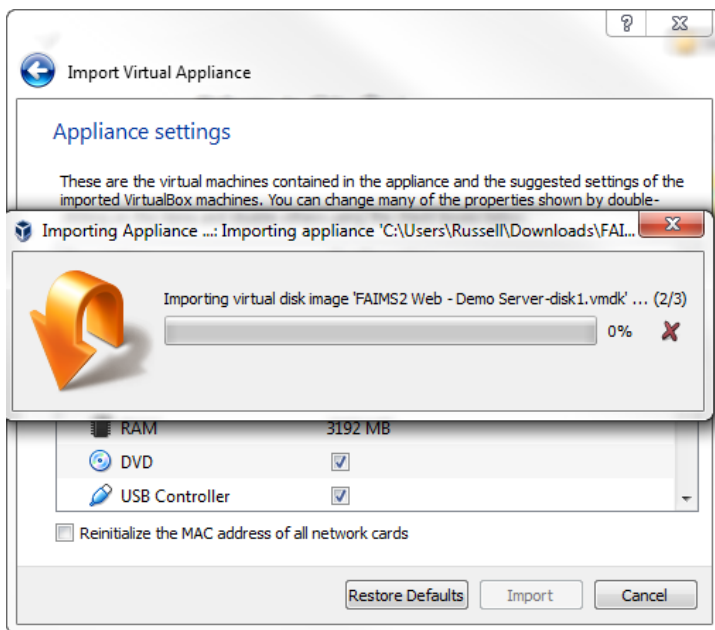


Figure 1.8

Once the setup is complete, "FAIMS2 Web - Demo Server" will appear in the menu on the left in VirtualBox.

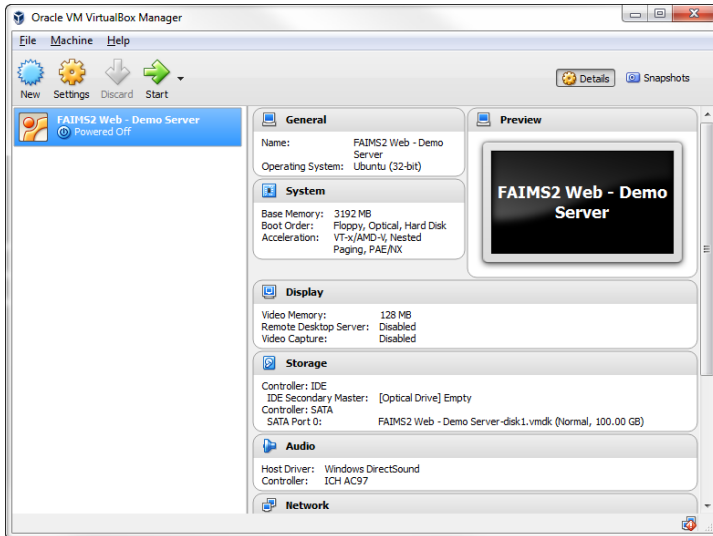


Figure 1.9

Click on the system and then click the “Start” button in VirtualBox to start the server. You *will* encounter the following error.

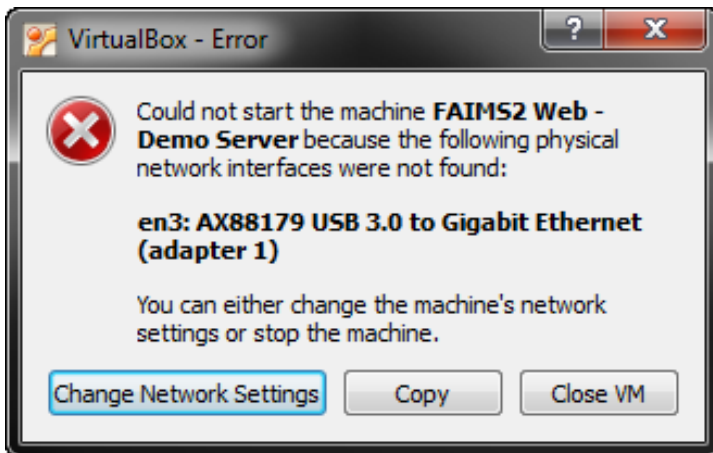


Figure 1.10

DON'T PANIC.

Simply, click the “Close VM” button. You’re still on the right track; sometimes, things just have to be a little fussy.

Back on the main screen, click on the “Network” section.

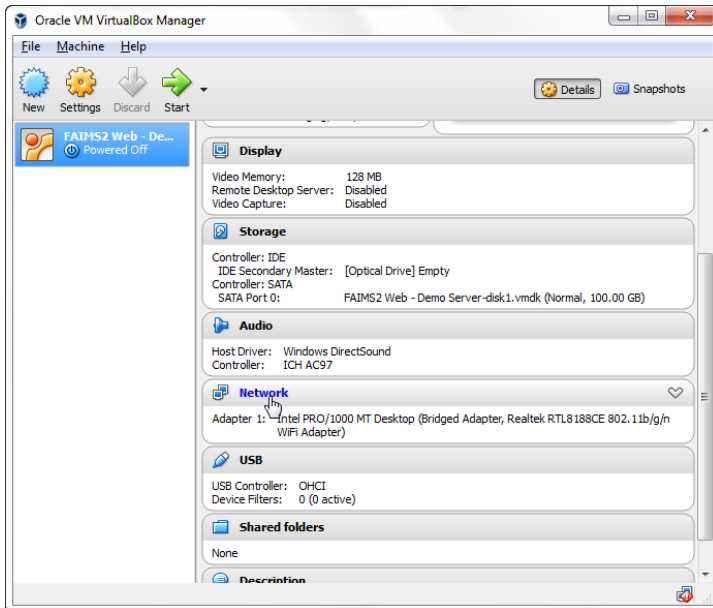


Figure 1.11

In the popup window that appears, check the “Enable Network Adapter” box on the “Adapter 1” tab, and select “Bridged Adapter” from the “Attached to:” menu.

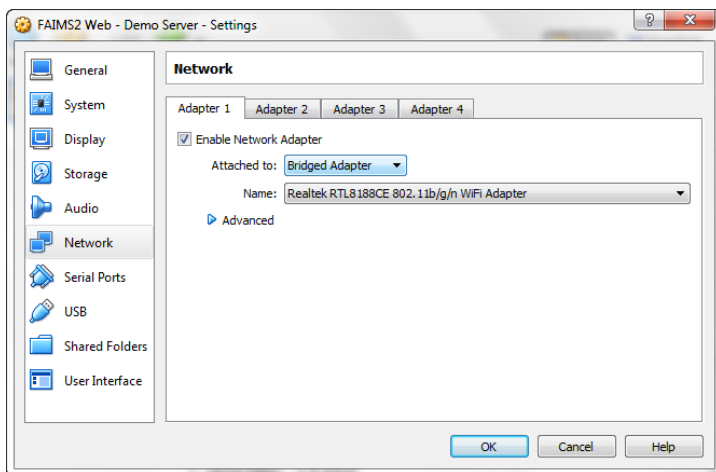


Figure 1.12

You may choose your machine's wifi adapter but, if you run into any issues, try using your hardline connection instead.

Click the “Start” button on the VirtualBox program. VirtualBox will pop up a new window in which your emulated machine will run. Give VirtualBox a few minutes to start your new emulated Ubuntu machine, and a few minutes longer for Ubuntu to start the FAIMS server. When they've finished, you will see the following message about the server having been set up.

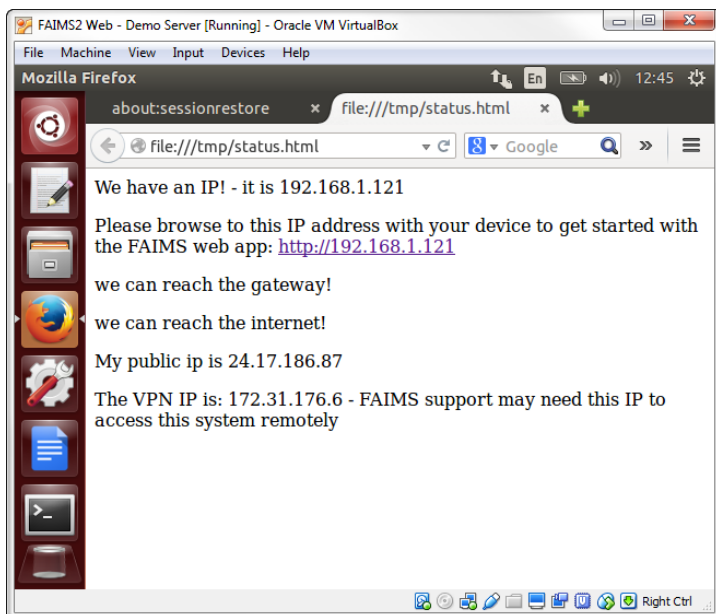


Figure 1.13

If you like, you can skip ahead a little and, back on your real machine, you can open the FAIMS server by navigating to 192.168.1.121, or whatever address is listed by your FAIMS status page in your web browser (check your messages for the specific IP address on your machine). **Note:** The default FAIMS account is "faimsadmin@intersect.org.au", password "Pass.123".

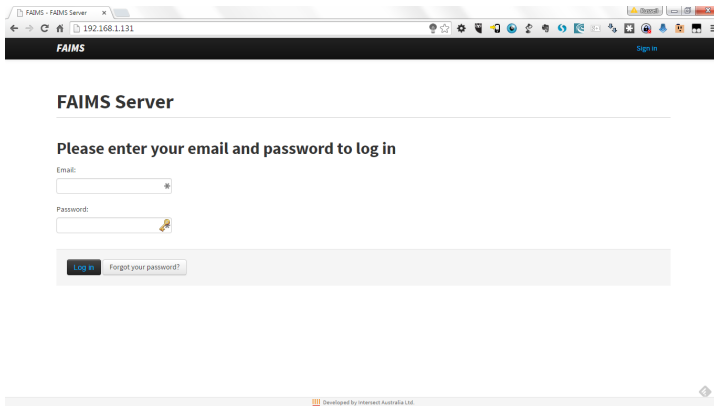


Figure 1.14

If you log out of the Ubuntu installation at any time, just use the password "Pass.123" to log back into the system.

1.1.6 Getting the FAIMS Server Up to Date

The final step for setting up the FAIMS server is to ensure it is up-to-date. To do this, inside Ubuntu, click on the Ubuntu symbol and type in "Terminal." Terminal (or “the terminal”) is a command line prompt that allows you to run many different programs using text commands. (To learn more about the Terminal and executing Bash commands, see here: https://help.ubuntu.com/community/UsingTheTerminal#In_Ubuntu).



Figure 1.15

Now, in the Terminal, type or copy the following commands individually (each line is a command) and press enter to run each of them:

```
$ wget http://fedarch.org/update.sh
$ bash ./update.sh
```

Code 1.1 Updating the server.

This may also be a good time to take a break. We like breaks.

1.2 Installing the FAIMS module “Auto-gen”

In addition to the FAIMS server, we’ll also install the FAIMS “auto-generator” (Autogen)³. The FAIMS server is what you’ll use to publish

³ The “Autogen” is a Domain Specific Language that we invented to make life easier for us to deploy modules. It turns out that this makes life easier for other people too.

Quiz 1.1 Test your knowledge about the VM

1. If the FAIMS server is supposed to be run on an Ubuntu machine, how are you going to run it on your non-Ubuntu computer?
2. What can you do if you can't find an option or setting we refer to in our instructions?
3. If you've installed VirtualBox before, what do you need to do before setting it up on your computer?
4. Before you import the server image into VirtualBox, what needs to be done to it?

and manage your module, but it's the scripts and programs that comprise the Autogen that will allow you to create your module's necessary files in the first place. It's the Autogen that allow someone to create all the necessary files of a module without significant programming experience.

1.2.1 Installing the prerequisites for the Autogen

The Autogen in FAIMS-Tools is built on top of and depends on a few other programs. Before we start using the Autogen, we'll need to make sure all of those programs are correctly installed and configured.

All of these tools are easier to install on a Linux system than on other operating systems, so we suggest downloading and installing them on the emulated Ubuntu installation you've just set up.

The programs the Autogen depends on are the XSLT processors *XSLT Proc* and *Saxonb-XSLT*. Some versions of Linux already have these programs installed. The Ubuntu image we instructed you to download,

however, does not. Fortunately, Linux has a handy command line program, apt-get, that provides access to tens of thousands of free software packages and makes installing programs like these straightforward⁴

To install our prerequisite tools, type or copy the command in [Code-block 1.2](#) and press enter to run it. Every charac:

```
$ sudo apt-get update && sudo apt-get install -y \ xsltproc gcj-4.3-jre-headless \
libsaxonb-java
```

Code 1.2 BASH shell commands to install xsltproc. We have to tell the apt-get program to look for the latest versions of files first, though.

This will probably take some time to install. Have more coffee.

You have now installed XSLT Proc, Java⁵, and libsaxonb. This program helps to transform the xml file you will be making into the data and ui schemas.

1.2.2 Installing the FAIMS Tools

To install the FAIMS-Tools, especially in Linux, use the version control software [Git](#). In the Terminal, on Linux and OSX, type "*git --version*" to see if Git is already installed. Type the following command into your terminal window on your Ubuntu installation:

```
git clone https://github.com/FAIMS/FAIMS-Tools.git
```

Git will copy all the files and notes in the GitHub repository (also known as a “repo”) to the directory from which you ran the Terminal command.

⁴ Brian’s note, straightforward, but a bit intimidating because it’s all knowledge-in-head instead of knowledge-at-hand. Power like this is... addicting though. Join us. We have cookies.

⁵ Brian’s note: Sorry.

1.2.3

1.2.4 Installing a Text Editor for Editing Necessary Files

The purpose of the FAIMS tools are to remove the need to individually craft all of your module's necessary files. Instead, you'll need to create one file the tools can use as a blueprint for making everything else. In order to make that one file (and to make any fancy edits to the necessary files to add detail or functionality), you're going to need a simple text editor that's good for coding with.

If you're doing your coding within the virtual machine, we recommend using an editor called gEdit which came with your Ubuntu installation. It's simple, functional, and won't clog up your instructions with unnecessary formatting like a standard Word Processor would.*

If you decide to write the files outside your virtual machine, we don't recommend using the text editors (such as Notepad) that come with Windows or OSX. While they may not complicate your code like a word processor would, they also don't have little features like autocomplete (a feature that means you don't need to fully type out words you use often) and syntax highlighting (a feature that helps you keep track of how your code is structured with helpful visual cues). With a little searching around, you can find plenty of excellent free or commercial text editors.

We particularly recommend:

- [*Notepad++*](#) (Windows)
- [*Atom*](#) (Cross Platform)
- [*TextWrangler*](#) (OSX)

For each of these, simply download their binary installers and follow their installation wizards. If you're using a Linux device that isn't the virtual machine, you can install gEdit, Vim, Emacs, or another text editor using apt-get.

****Note from your friends at FAIMS:*** One of the main reasons to use the text editors above is so that you can avoid "Smart Quotes," an automatic feature of many word processor programs that will insert differently styled or extra quotes that will cause the FAIMS server to reject your module. If you run into any issues with your module, ensure that "Smart Quotes" are disabled in your text editor.

1.2.5

1.2.6 Get Access to your local files via VirtualBox

1.2.7 Installing the Android SDK

Now we'll need to install the Android Software Development Kit (ADK). It's available at <https://developer.android.com/sdk/installing/index.html>. Ignore the Android Studio link; we just need the SDK Tools.

Because you're going to use the tools for debugging an Android device plugged into your actual computer, you're going to want to download and install SDK Tools to your regular computer system, NOT your Ubuntu virtual machine.

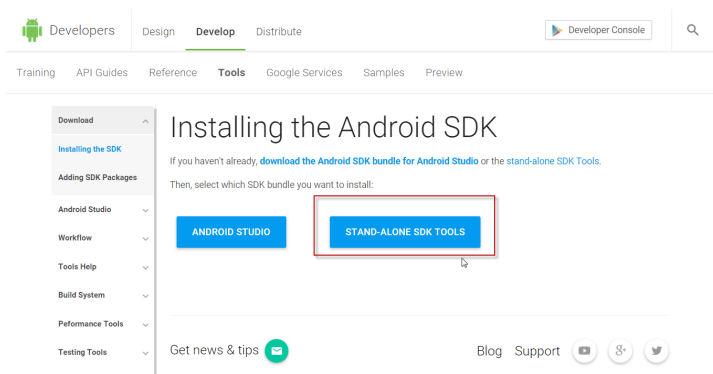


Figure 1.16

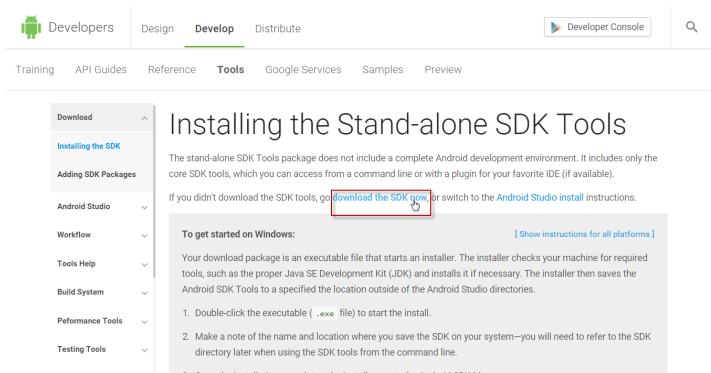


Figure 1.17

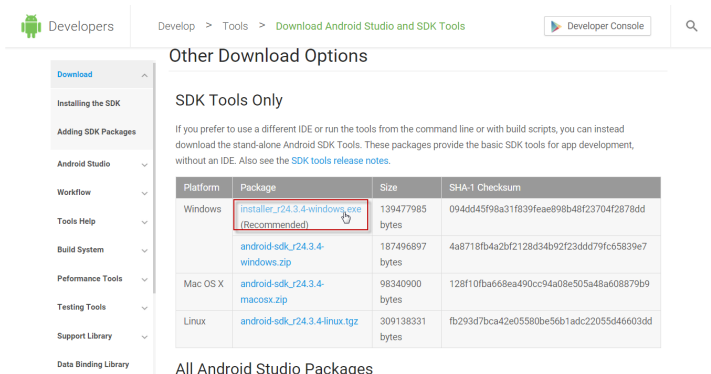


Figure 1.18

Run the downloaded installer and follow its prompts to finish setting up the SDK tools.

1.2.8 Installing the FAIMS Debug APK on your Android Device

The final step to setting up your development environment is to install the FAIMS app on your Android tablet or phone. Instead of installing the app through the Google Play Store, navigate to <https://www.fedarch.org/apk/debug-latest.apk> and download the special debug version.

Next, you'll need to place the FAIMS APK file on your phone, either by moving it over via USB connection or by placing the APK file in a cloud service like Dropbox and downloading it through the mobile app.

TIP: If you load the URL <https://www.fedarch.org/apk/faims-debug-latest.apk> on your Android devices, you'll be able to directly download the file.

To install the APK, you may need to change the security settings on your device to allow non-market apps. Go to the “Settings” app, find the Security settings, and check “Unknown sources” (device pictured is a Samsung Galaxy s5. Your settings menu may look different).

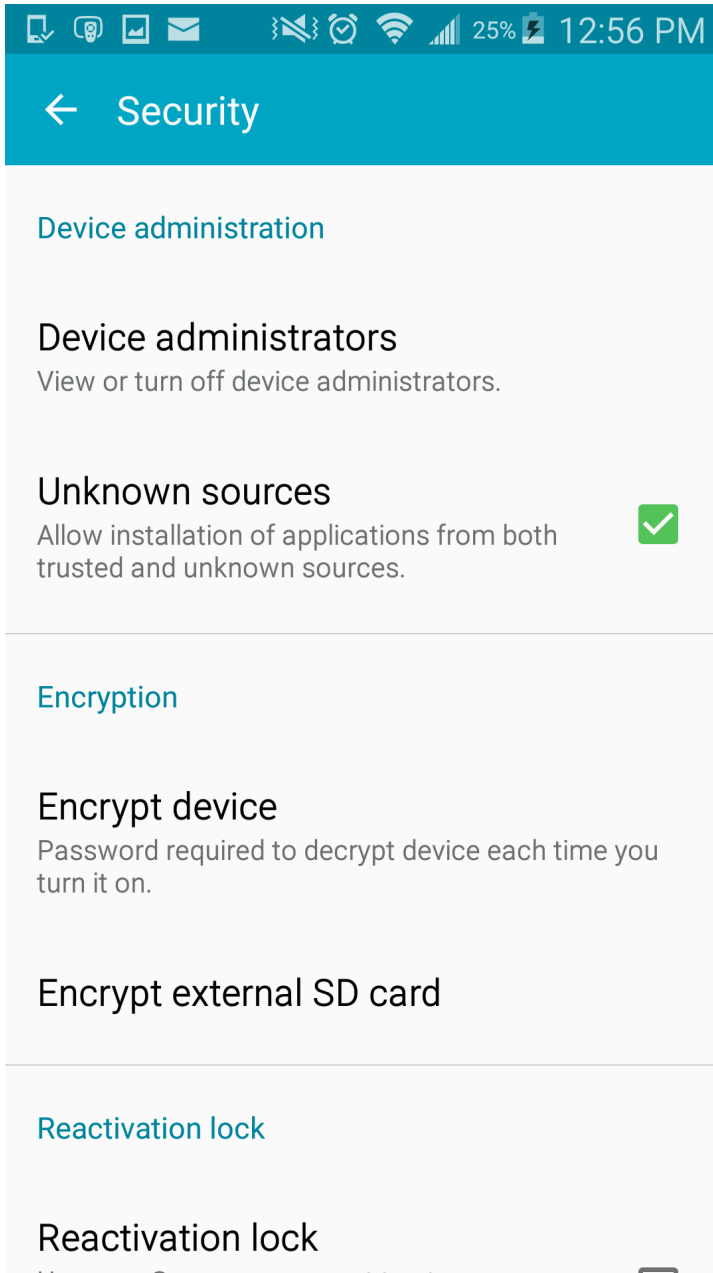


Figure 1.19

1.2.9 Making your Android Device Communicate with your Computer with USB Debugging

First, you'll need to enable the "Developer Options" on your device. This is another step where the instructions are different depending on what kind of device you're using. Below are instructions for enabling Developer Options with a Samsung Galaxy S5; if you have something different, you may need to Google your device and "enable developer options" together.

Note that if you want for USB Debugging to work when your Galaxy S5 is connected with your computer, then you will first need to make sure that the [Samsung USB Drivers are installed on your PC](#).

<http://www.android.gs/download-samsung-usb-drivers-for-android/>

For many non-Samsung devices, you can use Google's ABD/USB driver instead: <https://developer.android.com/studio/run/oem-usb.html#InstallingDriver>

To enable the "Developer Options":

1. Open the App drawer.
2. Launch Settings menu.
3. Find the open the 'About Device' menu.
4. Scroll down to 'Build Number'.
5. Next, tap on the 'Build Number' section seven times. (Yes, really.)
6. After the seventh tap you will be told that you are now a developer. (Again, this is really how it works.)
7. Go back to Settings menu and the Developer Options menu will now be displayed.

In order to enable the USB Debugging you will simply need to open Developer Options, scroll down and tick the box that says 'USB Debugging'.

That's it! Your FAIMS development environment should now be set up properly.

1.2.10 Test Your Knowledge

What are two things you can do if you don't find an option or setting where we tell you it will be?

If the "Error: Network Device Not Found" error appears when you start VirtualBox, how do you fix it?

True or False: Microsoft Word is a good program to code your modules in.

Should you install your Android developer kit within the virtual machine? Why or why not?

Which of the following are true?

- VirtualBox allows you to run a different operating system on your computer
- VirtualBox connects to a computer at the FAIMS offices
- You must install XSLT Proc and saxon-xslt before running the XML generator tool
- FAIMS can use both Android and Apple devices

Exercise: Open up a command window (Windows key + type "cmd" + hit enter/select with mouse. In this command window, enter the text "adb -version" then enter. Some information about the Android Development Bridge (ADB) program should appear. Did you encounter any errors? If not, congratulations, ADB/Android SDK is successfully installed.

1.3 How to Code Modules