



NEXT GENERATION FIRST RESPONDER HOW-TO GUIDE

Configuring a Raspberry Pi Location Sensor

EXECUTIVE SUMMARY

The Department of Homeland Security (DHS) Science and Technology Directorate (S&T) launched the Next Generation First Responder (NGFR) Apex program in January 2015 as a strategic initiative to develop and integrate next-generation technologies with the goal of expanding first responder mission effectiveness and safety. The NGFR Apex program seeks to help tomorrow's first responder be better protected, connected and fully aware.

NGFR has conducted three integration demonstrations to assess how prototype technologies integrate to support the NGFR Apex program. The first demonstration in January 2016 was primarily a tabletop demonstration, and the second in May 2016 was a combined tabletop and field demonstration in a large urban area. For the third NGFR integration demonstration, the NGFR Apex program wanted to collaborate with a more rural community where cutting-edge technologies face a unique set of deployment challenges.

NGFR partnered with first responders in Grant County, Washington, to assess the capabilities of several NGFR technologies to support their public safety operations. This effort, known as the Grant County – DHS Science and Technology Directorate Next Generation First Responder Apex Program Technology Experiment (TechEx), involved deploying a suite of technologies. This guide presents a setup guide for an application (PiPoint) that can be used by public safety agencies as an example of how to set up a low-cost single board computer to provide location services via Wi-Fi or LTE.

Initial discussions with Grant County identified the following five areas as priority needs:

- Geo-location of first responder vehicles and personnel on map displays at the Grant County Multi-Agency Communications Center (MACC), command posts and on smartphones.
- Wireless data service at the Gorge Amphitheatre concert venue, campgrounds and along the Columbia River valley using various broadband technologies, including cellular broadband (Long Term Evolution [LTE]), Wi-Fi and digital television datacasting.
- Ability to view real-time video at the MACC, command posts, emergency management centers and other destinations when captured and streamed from small Unmanned Aircraft Systems (sUAS) or first responder smartphones.
- Capability to monitor first responders' physiological conditions and send the data wirelessly to the MACC and command post(s) for viewing using a visual "dashboard" on a monitor.
- Support for communications and information dissemination using a combination of county-owned land mobile radios (800 MHz P25), commercial mobile networks and a deployable government-band public safety broadband network (Band 14 LTE) for data communications.

After further discussions, extensive planning, site visits, an integration and testing event, and a dry run of the experiment, NGFR and Grant County conducted the TechEx on June 6 and 7, 2017. The TechEx involved the deployment of more than 50 first responders from multiple Grant County first responder agencies, as well as 15 S&T NGFR staff and support contractors.

What's Inside?

- 1.....Executive Summary
- 3.....Introduction
- 4.....TechEx Overview
- 11.....Results
- 14.....Implementation for Your Agency
- 15.....Summary
- 16.....References & Recommended Reading

Want More?

To learn more about the NGFR Apex program, associated projects, and how DHS supports first responders nationwide, tune into our website and social media accounts and LIKE, CLICK and SHARE!



WEBSITE

<http://www.dhs.gov/NGFR>



FACEBOOK

[@DHSSciTech](#)



TWITTER

[@DHSSciTech](#)



EMAIL

First.Responder@hq.dhs.gov



Homeland Security

Science and Technology

INTRODUCTION

Next Generation First Responder Apex Program

The Department of Homeland Security (DHS) [Science and Technology Directorate](#) (S&T) launched the [Next Generation First Responder \(NGFR\) Apex program](#) in January 2015 as a strategic initiative to develop and integrate next-generation technologies with the goal of expanding first responder mission effectiveness and safety. The NGFR Apex program seeks to help tomorrow's first responder be better protected, connected and fully aware. When firefighters, law enforcement officers and emergency medical services have enhanced protection, resilient communications and advanced situational awareness, they are better able to protect our communities and make it home safely. The NGFR Apex program develops, adapts and integrates cutting-edge technologies using open standards, increasing competition in the first responder technology marketplace and giving responders more options to build the systems they need for their mission and budget.



Purpose of This How-To Guide

This how-to guide provides installation and configuration instructions of the PiPoint location software for a Raspberry Pi computer that was demonstrated as part of the Spiral 2 technology exercise. This guide does not document all of the components required to build an operational system.

It is recommended this guide is used by an IT or engineering professional that has familiarity with the installation, configuration and testing of hardware and software. Some familiarity with Linux concepts is needed for using this guide. This guide is not written with a casual technology user in mind.

There are three other external software/hardware components that are necessary to render the Raspberry PiPoint location sensor functional. These are listed below.

A message router, which can be found at <https://github.com/1stResponder/fresh>. This is software that will distribute data received from the Raspberry Pi location device. It will need to be installed and configured to allow location data to be distributed to data consumers.

A router / GeoDB package will need to be installed to convert messages received from the Raspberry Pi location device and FRESH router into a format that can be used by Geographic Information Systems (GIS). The package can be found at <https://github.com/1stResponder/router-geodb>.

Some form of Situational Awareness application will need to be available to display the “dots on the map” picture. The situational tool that will interoperate with the other components identified above is located at <https://github.com/1stResponder/watchtower>.

There are other implementations of this capability available online, as well using other open source models. The details of implementation will have to be based on an individual agency's needs and existing infrastructure.

The components for an alternative approach used in another Spiral demonstration is described in https://www.dhs.gov/sites/default/files/publications/IoT_Extension_Report_OGC_final-508.pdf. The description of items can be found in section 7.4.2.5. This document describes a different approach to the implementation of sensor devices.

<http://docs.opengeospatial.org/per/16-092r2.html> is also a description of the earlier implementation Spiral demonstration.

HOW TO CONFIGURE A RASPBERRY PI LOCATION SENSOR

The purpose of this how-to guide is to describe how to configure a Raspberry Pi single board computer (SBC) to be able to provide location tracking capabilities using low-cost components and PiPoint software.

Prerequisites

1. Access to a computer (this guide gives the instructions for Mac OSX and Windows) with network capabilities.
2. A computer monitor to plug the Raspberry Pi into.
3. Wi-Fi access for the **Raspberry Pi**.
4. The hardware listed below.



Hardware

1. A Raspberry Pi 3 with built-in Wi-Fi capabilities and a power supply (the Raspberry Pi uses a microUSB cable).
2. A microSD card with at least 8GB of storage.
3. An adapter to plug the microSD card into a computer.
4. The GlobalSat BU-353S4 USB GPS Receiver.
5. A USB mouse and keyboard.
6. An HDMI cable and a monitor to are required to connect to the Raspberry Pi.
7. We used the CanaKit starter pack, which includes the Pi (with built-in Wi-Fi & Bluetooth), the case, a 32GB microSD card, a USB microSD card reader, the power supply and an HDMI cord.

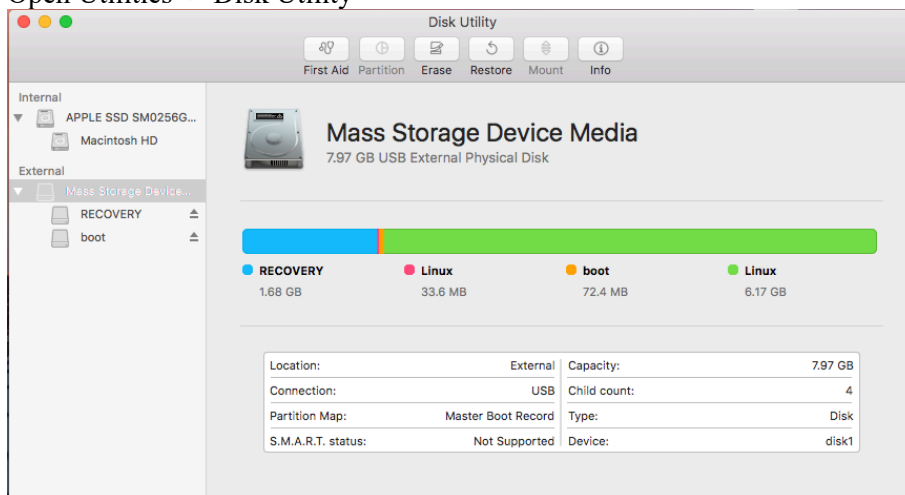
Raspberry Pi SBC Operation System Setup

If the microSD card did not come preloaded with the New Out Of the Box Software (NOOBS), you will need to install it. Use the following steps if the Micro SD card is not preconfigured. Otherwise, go to the “Install Raspbian” section.

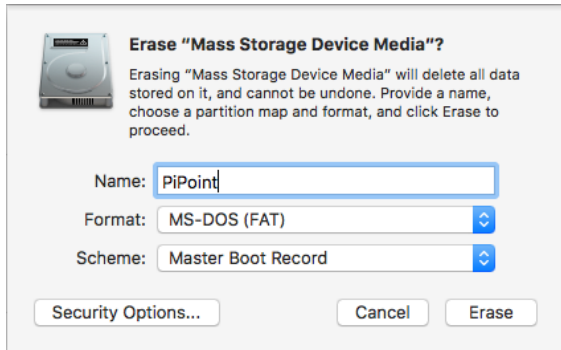
Format MicroSD Card

Mac OS

1. Plug the microSD card into your machine
 - a. Do not access the card through the Pi
2. Open Utilities -> Disk Utility



3. Select your microSD card from the left-hand side
4. Click “Erase” on the top bar
5. Select “MS-DOS (FAT)” under format (default scheme is fine)

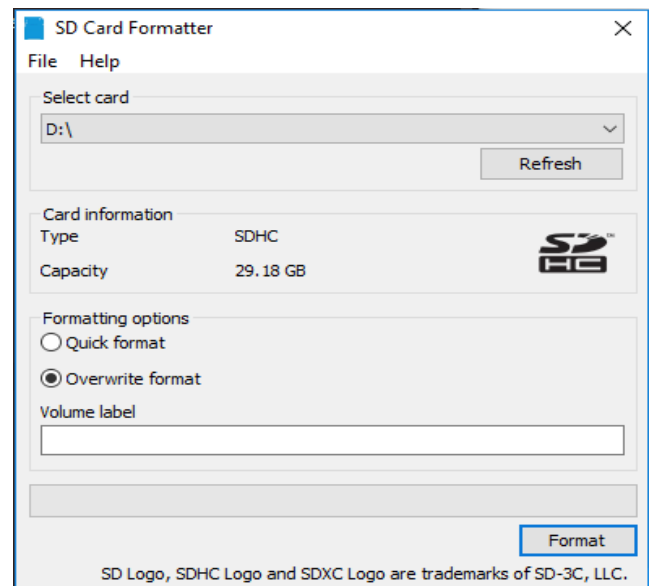


6. Click Erase
7. Your microSD card will now be properly formatted for Raspbian

Windows

The recommended formatting tool to use is the SD Associations Formatting Tool.

1. Plug the microSD card into your computer
 - a. Do not access the card through the Pi
2. Download the Window’s version of the SD Association’s formatting tool from: https://www.sdcard.org/downloads/formatter_4/
3. Run the installer
4. Launch the program
5. Select your microSD card from the list of choices
6. Select “Overwrite format” under formatting options
7. Click “Format”
8. Your microSD card will now be properly formatted for Raspbian



Install NOOBS

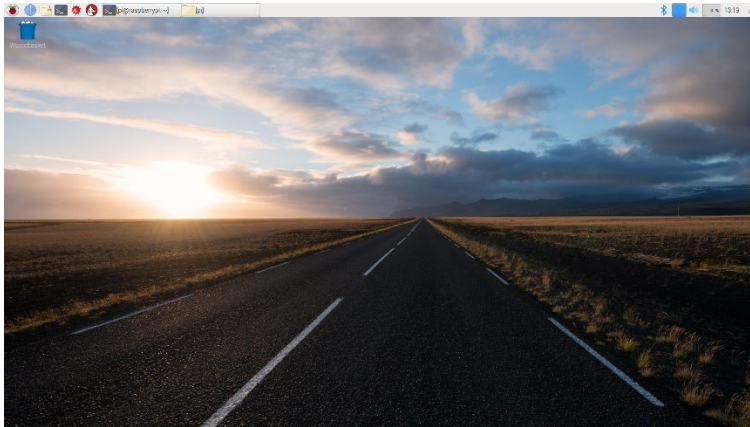
1. Download NOOBS from: <https://www.raspberrypi.org/downloads/>
2. Extract the files and copy them to the microSD card
 - a. Make sure the files are placed directly in the root of the microSD card. Extracting files can sometimes result in them being placed in a folder, but you do not want this folder on the microSD card; just the files themselves.

Install Raspbian

Raspbian is the official operating system for the Raspberry Pi.

1. Ensure that the Raspberry Pi is connected to an external monitor, keyboard and mouse
2. Unplug the microSD card from your computer and insert it into the Raspberry Pi
3. Boot up the Raspberry Pi
4. You should see the NOOBS utility

5. Select Raspbian (not Raspbian Lite) from the list
6. Click “Install”
7. Once the installer is finished, Raspbian should automatically boot up. You should see the Raspbian desktop:



8. You can now shut off the Raspberry Pi via menu selection

Raspberry Pi 3 Serial Port Configuration Instructions

PI 3

1. Take the microSD back from the Raspberry Pi and plug it into your computer
2. Open “config.txt” under the boot partition
3. Add the following line to the end of the file:

```
enable_uart=1
```

4. Close and save
5. Remove the microSD card from your computer once again and place it back in the Raspberry Pi
6. Boot up the Pi and open the terminal window
7. Run the following commands:

```
sudo systemctl stop serial-getty@ttyS0.service
```

```
sudo systemctl disable serial-getty@ttyS0.service
```

8. Continue on to PiPoint Setup

PiPoint Setup

The Raspberry Pi is now all ready for the PiPoint installation.

Install Packages

Make sure the Raspberry Pi is powered on and connected to the internet via Wi-Fi.

Dependencies

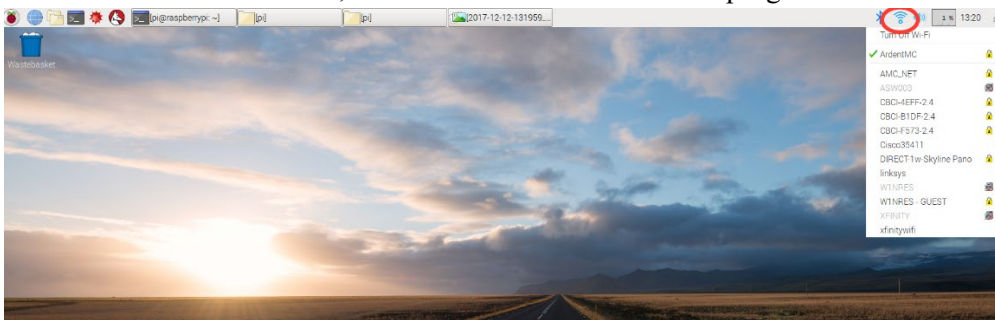
PiPoint is dependent upon the following packages:

- Python 2 (should come preloaded)
- GPSD
- GPSD-Clients
- Python-gps
- Python-pip
- dnsutils
- vim
- supervisor
- git
- ntp
- ntpdate

Except for Python 2, they are loaded in the following steps.

Install Packages

1. Make sure the Raspberry Pi Wi-Fi is turned “ON” and it is connected to a network with internet access
 - a. To connect to Wi-Fi, click the Wi-Fi icon from the top right corner of the screen and select a network



2. Open a terminal window on the Raspberry Pi
3. Run the following commands:

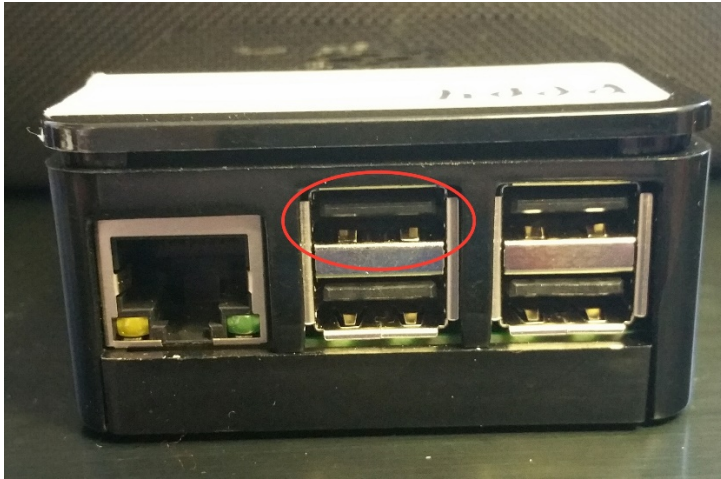
```
sudo apt-get update
sudo apt-get install gpsd gpsd-clients python-gps python-pip dnsutils vim insserv
supervisor git ntp ntpdate
```

4. Accept any prompts that pop up during install

GPS Setup

If the GPS is in the wrong slot or not working, then PiPoint will not be able to report position.

1. Make sure the GPS is plugged into the usb0 slot of the raspberry Pi (see photo below)



2. Run the following command on the Raspberry Pi terminal:

```
stty -F /dev/ttyUSB0 ispeed 4800 && cat </dev/ttyUSB0
```

3. You should see the output of the GPS. If you get an error or do not see any output, try moving the GPS puck or unplugging and re-plugging in the GPS puck.

Install PiPoint

Download

1. With web browser, obtain code located at: <https://github.com/1stResponder/pi-loc>
2. Download and extract the PiPoint code
3. Open the terminal window if it is not already open
4. Create a directory called “pi-loc” inside /opt

```
sudo mkdir /opt/pi-loc
```

5. Navigate to where you placed the PiPoint code on the Pi via the terminal
6. Copy the contents of the src directory into: /opt/pi-loc

```
sudo cp -a src/ /opt/pi-loc/
```

7. Copy the conf directory into pi-loc

```
sudo cp -r conf /opt/pi-loc/
```

Configuration File

The configuration file can be found at /opt/pi-loc and is called config.cfg. The configuration file contains data elements needed for the EDXL/DE data format and the elements that are required to make the data usable.

```
[Main]
UnitPrefix: VA.LCFR # The Unit Prefix
UnitID: DC999 # The UnitID
Sender: CAD@loudoun.gov # The sender ID
```

Type: ATOM_GRDTRK_EQT_GRDVEH_CVLVEH_EM_FIRE_ENGINE	# The Resource Type
ContentDesc: LCFR	# The Content Description
Keyword: LCFR Resource	# The DE Content Keyword
Stale: 30 # in minutes	# Amount of time message is considered valid
URL: http://fresh-test.psccloud.org/api/de	# The URL to Post to
Logging: DEBUG	# Logging Level
LogLocation: /var/log	# Log Location

Installation

If you wish to change any settings on PiPoint or want to place the code in another location, you will need to update the package configuration file.

1. Navigate to /opt/pi-loc/conf via the terminal
2. Open “pi-loc.conf” in editor of your choice
3. Make necessary changes
4. Now copy this file into: /etc/supervisor/conf.d/

```
sudo cp pi-loc.conf /etc/supervisor/conf.d/
```

5. Next, copy the “gpsd” file into etc/default

```
sudo cp gpsd /etc/default
```

a. You will be overwriting the file that is already there

6. Then copy “set-rofs” and “set-rwfs” into /usr/local/sbin

```
sudo cp set-rofs /usr/local/sbin
```

```
sudo cp set-rwfs /usr/local/sbin
```

7. Chmod the files to be 744:

```
sudo chmod 744 /usr/local/sbin/set-rwfs
```

```
sudo chmod 744 /usr/local/sbin/set-rofs
```

8. Now run the commands:

```
update-rc.d supervisor defaults
```

i. If asked for authentication, the default username is “pi” and the default password is “raspberrry”

```
pip install requests
```

9. Once the commands have finished, restart the Raspberry Pi
10. Your PiPoint should automatically begin reporting information

Confirm Supervisor is Running

1. Run the following command in the terminal window:

```
sudo service supervisor status
```

Confirm PiPoint is Working

You can confirm PiPoint is working by checking the log files.

1. Navigate to: /var/log
2. Confirm that the “pi-loc.out” file exists and is still being written to

Configure Raspberry Pi For Read-only

It is highly recommended that you create an image backup of your Raspberry Pi before continuing on to these next steps. See [Create Backup](#).

Create Read-Only FS

Clean Up

You will need to upgrade the necessary packages and remove any unneeded packages/services before making the Raspberry Pi read-only.

1. Update the packages with the following commands in the terminal window:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

2. Remove the unneeded packages and x11 with the following commands in the terminal window:

```
sudo apt-get remove --purge wolfram-engine triggerhappy anacron logrotate dphys-swapfile xserver-common lightdm
```

```
sudo insserv -r x11-common
```

```
sudo apt-get autoremove --purge
```

- a. At this point, a lot of the visuals and extras bundled with Raspbian will be removed

Configure Logging

This will put log into circular memory buffer, which allows us to see the logs using the logread command.

1. Replace default log management with busybox using the following command in the terminal window:

```
sudo apt-get install busybox-syslogd
```

```
sudo dpkg --purge rsyslog
```

Configure System

Disable Swap and Filesystem Checks

1. Navigate to: /boot
2. Modify “cmdline.txt” by adding the following to the end of the line in the terminal window:

```
fastboot noswap ro
```

a. e.g.:

```
dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1 root=PARTUUID=7dedfa85-02  
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait fastboot noswap ro
```

Move Files to The TMPFS Filesystem

1. Navigate to the root directory in the terminal window
2. Run the commands:

```
sudo rm -rf /var/lib/dhcp/ /var/run /var/spool /var/lock /etc/resolv.conf
```

```
sudo ln -s /tmp /var/lib/dhcp
```

```
sudo ln -s /tmp /var/run
```

```
sudo ln -s /tmp /var/spool
```

```
sudo ln -s /tmp /var/lock
```

```
sudo touch /tmp/dhcpd.resolv.conf
```

```
sudo ln -s /tmp/dhcpd.resolv.conf /etc/resolv.conf
```

3. Navigate to /etc/systemd/system
4. Modify the file “dhcpd5.service” and replace the value for “PIDFile”:

```
PIDFile=/var/run/dhcpd.pid
```

5. Restart DHCP with the command in the terminal window:

```
sudo systemctl daemon-reload
```

```
sudo service dhcpd restart
```

Move Random-Seed

1. Navigate to the root directory
2. Run the commands:

```
sudo rm /var/lib/systemd/random-seed
sudo ln -s /tmp/random-seed /var/lib/systemd/random-seed
```

Create The Target At Boot-Time

1. Navigate to /lib/systemd/system
2. Modify the file “systemd-random-seed.service” by adding this line to the “[Service]” section:

```
ExecStartPre=/bin/echo "" >/tmp/random-seed
```

3. Restart system with the following command:

```
sudo systemctl daemon-reload
```

Confirm the Pi Using NTP and the Time Zone is Set

1. Run the following to confirm ntpq is running:

```
sudo ntpq -p
```

2. Now configure the time zone for ntpq by running:

```
sudo raspi-config
```

3. When the Raspberry Pi Configuration Tool boots up, select “4 Localization Options”
4. Then “Change Timezone”
5. Select the desired time zone
6. Then "Finish”

Configure Clock to Properly Handle FORS

1. Navigate to /etc/cron.hourly
2. Modify the file “fake-hw-clock”, and replace its contents with:

```
if (command -v fake-hwclock >/dev/null 2>&1); then
mount -o remount,rw /
fake-hwclock save
mount -o remount,ro /
fi
```

Set NTP to put the drift file in TMPFS

1. Navigate to /etc
2. Modify the file “ntp.conf”. Change the value of “driftfile” to be:

```
driftfile /var/tmp/ntp.drift
```

Remove Unneeded Startup Scripts

1. Run the command:

```
sudo insserv -r bootlogs; insserv -r console-setup
```

- i. If the services are already deleted, that is fine

Create TMPFS Mounts

Get the PARTUUIDs

First, you will need to get the PARTUUIDs for your system drives.

1. Run the following command to and note the PARTUUID for disk 1:

```
sudo blkid /dev/mmcblk0p1
```

2. Then repeat for disk 2:

```
sudo blkid /dev/mmcblk0p1
```

Modify FSTAB

1. Navigate to: /etc
2. Replace the file contents of “fstab” with the following:

```
proc          /proc        proc defaults      0 0
PARTUUID=<UUID_DSK1> /boot        vfat defaults,ro    0 2
PARTUUID=<UUID_DSK2> /            ext4 defaults,noatime,ro 0 1
tmpfs         /tmp         tmpfs nosuid,nodev     0 0
tmpfs         /var/log    tmpfs nosuid,nodev     0 0
tmpfs         /var/tmp    tmpfs nosuid,nodev     0 0
```

- a. Substitute < UUID_DSKX > for the PARTUUIDs found earlier

3. Navigate to: /etc/supervisor/
4. Modify the file “supervisor.conf” to be:

```
logfile=/var/log/supervisord.log ; (main log file;default $CWD/supervisord.log)
childlogdir=/var/log/          ; ('AUTO' child log dir, default $TEMP)
```

5. Determine which version of Raspbian you are using (stretch or jessie). Run the following command:

```
cat /etc/debian_version
```

- a. If the results start with >= 9, then it is stretch. If it is 8, then it is jessie.
6. Navigate to the lib/systemd folder:
 - a. If using jessie, that is at: /usr/lib/systemd/system
 - b. If using stretch, that is at: /lib/systemd/system
7. Modify the file “systemd-hostnamed.service” so the value of “PrivateTmp” is:

```
PrivateTmp=no
```

8. Reboot the Pi
 - a. To shutdown, run the command:

```
sudo shutdown -h now
```

Confirm Everything Works as Expected

1. Confirm the drives mounted correctly by running the command:

```
mount
```

- a. You should see two mmcblk partitions listed (the actual number of the partition does not matter). Make sure they match the following:

```
/dev/mmcblk0p2 on / type ext4 (ro,noatime,data=ordered)
```

```
/dev/mmcblk0p1 on /boot type vfat  
(ro,relatime,fmask=0022,dmask=0022,codepage=437,ioccharset=ascii,shortname=mixed,errors=remount-ro)
```

- b. Make sure that the ro (read-only) property is there
2. Make sure supervisor is running and healthy by running the following:

```
sudo service supervisor status
```

3. Check the pi-loc logs at: /var/log

Operating PiPoint

Once installed, the PiPoint requires no further configuration or steps to report information. Just plug the Pi into a power supply with the GPS puck and it will start running automatically.

Switch PiPoint to Read/Write

1. Run the command

```
sudo mount -o remount,rw /
```

Switch PiPoint to Read-Only

Before deploying PiPoint to the field, it should be set to Read-Only. This helps make the Pi more reliable.

1. Run the command

```
sudo mount -o remount,ro /
```

Change Configuration

You are able to modify the app settings (including logging and post URL) from the config file. Before attempting to make any configuration changes, make sure that the Pi is in read/write mode and not read-only: see [Switch Pi-Point to Read/Write](#).

1. From the Raspberry Pi terminal, navigate to the install location of PiPoint
2. Open the “config.cfg” file in VIM

3. Make any necessary changes

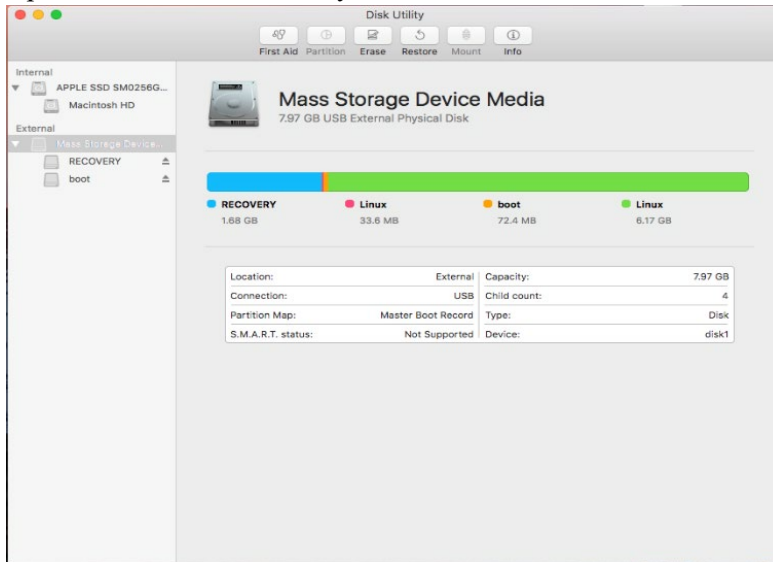
Backup Image

Creating a backup image of the Raspberry Pi SD card is good practice and prevents work being lost in case something goes wrong.

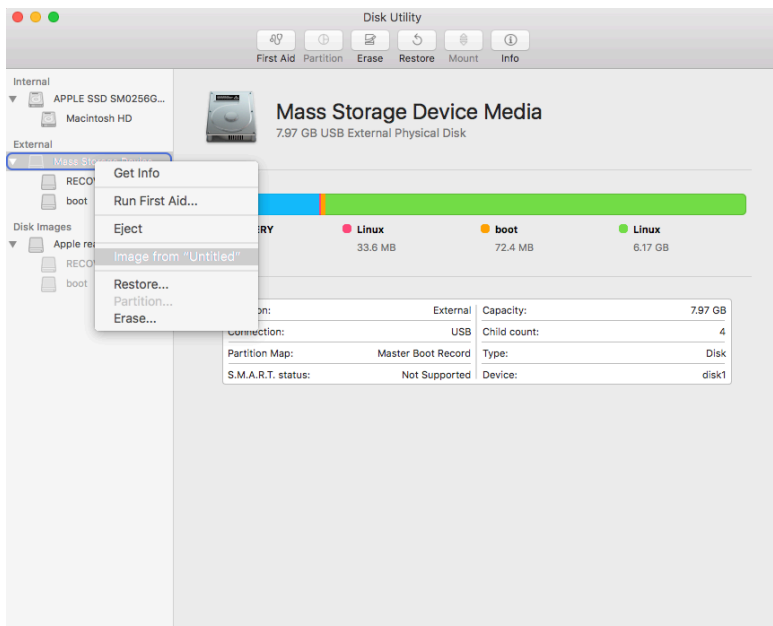
Create Backup Image

Mac OS

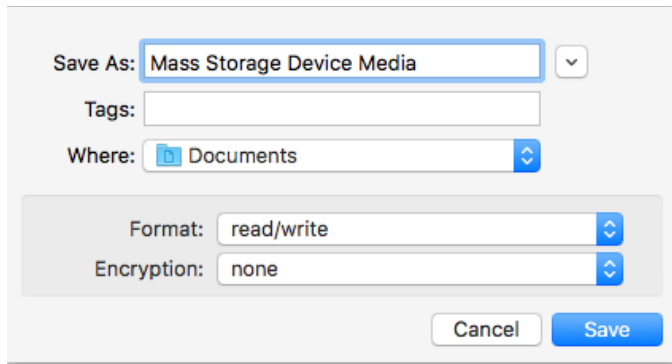
1. Take the microSD card from the Raspberry Pi and insert it into your computer
2. Open Utilities -> Disk Utility



3. Select your microSD card from the left-hand side
4. Right click your microSD card from the left-hand side



5. Select "Image From..."

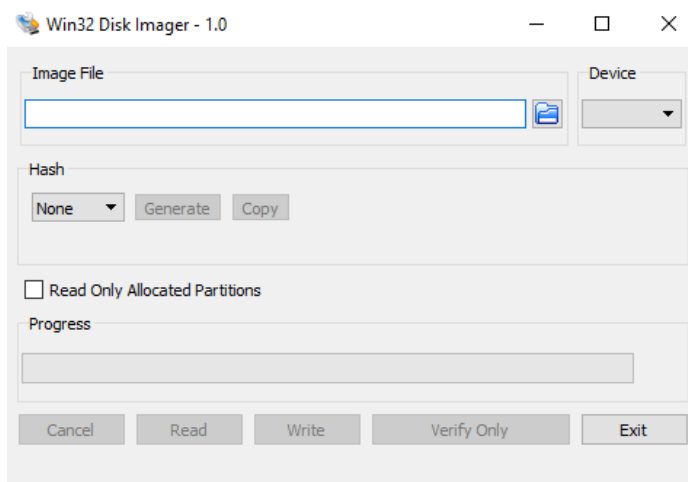


6. Select “Read/Write” for format
7. Click “Save”
 - a. Note: this will take a while to run

Windows

The recommended approach is to use Win32 Disk Imager.

1. Take the microSD card from the Raspberry Pi and insert it into your computer
2. If not already installed, download and install Win32 Disk Imager from: <https://sourceforge.net/projects/win32diskimager/files/latest/download>
3. Start up the program
 - a. Make sure the SD card is inserted before starting!



4. Select the microSD card under the device dropdown
5. Specify a filename and install location
6. Click “Read” to back up the microSD card to a location of your choice

Restore From Backup Image

Mac OS

1. Plug the microSD card into your computer
2. Open the terminal and run the following command:

```
diskutil list
```

3. Find your SD card from the list of given disks

4. Unmount your SD card with the following command:

```
diskutil unmountDisk /dev/<DISK_NAME>
```

5. Write the image back to the SD card:

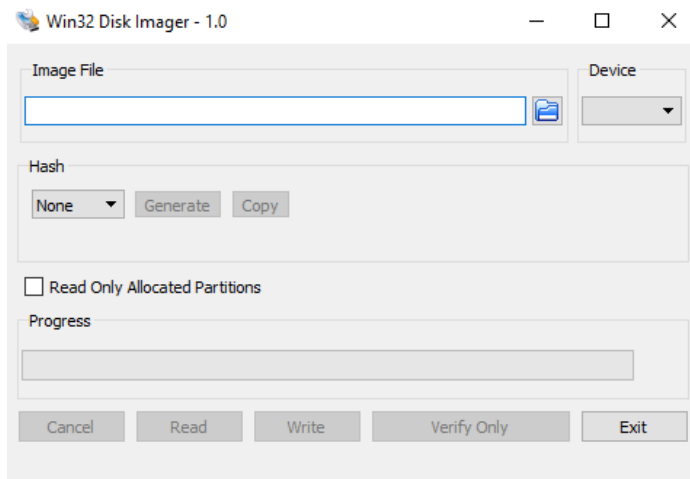
```
sudo dd if=~/<BACKUP_FILE>.dmg of=/dev/<DISK_NAME>
```

- a. NOTE: This command does not give any feedback until it is finished and can take a good amount of time. Do not unplug the microSD card or interrupt this process.

Windows

The recommended approach is to use Win32 Disk Imager.

1. Plug the microSD card into your computer
2. If not already installed, download and install Win32 Disk Imager from: <https://sourceforge.net/projects/win32diskimager/files/latest/download>
3. Start up the program
 - a. Make sure the SD card is inserted before starting!



4. Select the backup image using the blue folder icon
5. Select the microSD card under the device dropdown
6. Click "Write" to restore the microSD card from the image

Message Format

PiPoint uses the EDXL/NIEM exchange to transmit messages as HTTP Rest Post requests.

Protocol

PiPoint transmits messages over HTTP/HTTPS REST as post requests.

EDXL DE

PiPoint sends EDXL (Emergency Data Exchange Language) DE messages with a NIEM payload. An example message is shown below:


```

<EDXLDistribution xmlns="urn:oasis:names:tc:emergency:EDXL:DE:1.0"
  xmlns:emevent="http://release.niem.gov/niem/domains/emergencyManagement/3.1/emevent/0.1/"
  xmlns:emlc="http://release.niem.gov/niem/domains/emergencyManagement/3.1/emevent/0.1/emlc/"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:mof="http://release.niem.gov/niem/domains/militaryOperations/3.2/"
  xmlns:nc="http://release.niem.gov/niem/niem-core/3.0/">
  <distributionID>PI.LOC.PI999</distributionID>
  <senderID>0000000c3fbbab5@pipoint/senderID>
  <dateTimeSent>2017-11-27T20:23:28Z</dateTimeSent>
  <distributionStatus>Actual</distributionStatus>
  <distributionType>Update</distributionType>
  <combinedConfidentiality>U</combinedConfidentiality>
  <language>en-US</language>
  <distributionReference>PI.LOC.PI999,0000000c3fbbab5@pipoint,1753-01-01T00:00:00.0000000Z</distributionReference>
  <contentObject>
    <contentDescription>PILOC</contentDescription>
    <contentKeyword>
      <valueListUrn>http://edxlsharp.codeplex.com/ValueLists/ContentKeywords</valueListUrn>
      <value>PILOC</value>
      <value>PI.LOC.PI999</value>
      <value>ATOM</value>
    </contentKeyword>
    <xmlContent>
      <embeddedXMLContent>
        <emlc:Event>
          <mof:EventID>
            <nc:IdentificationID>PI.LOC.PI999</nc:IdentificationID>
          </mof:EventID>
          <emevent:EventTypeDescriptor>
            <emevent:EventTypeCode>ATOM</emevent:EventTypeCode>
            <emevent:EventTypeDescriptorExtension>ATOM</emevent:EventTypeDescriptorExtension>
            <emevent:EventTypeDescriptorExtension>RE</emevent:EventTypeDescriptorExtension>
          </emevent:EventTypeDescriptor>
          <mof:EventLocation>
            <mof:LocationCylinder>
              <mof:LocationPoint>
                <gml:Point SrsName="http://metadata.ces.mil/mdr/ns/GSIP/crs/WGS84E_3D">
                  <gml:pos>38.94896047 -77.334700266</gml:pos>
                </gml:Point>
              </mof:LocationPoint>
              <mof:LocationCylinderRadiusValue>-99999</mof:LocationCylinderRadiusValue>
              <mof:LocationCylinderHalfHeightValue>-99999</mof:LocationCylinderHalfHeightValue>
              <mof:LocationCreationCode>HUMAN</mof:LocationCreationCode>
            </mof:LocationCylinder>
          </mof:EventLocation>
          <mof:EventValidityDateTimeRange>
            <nc:StartDate>
              <nc:DateTime>2017-11-27T20:23:28Z</nc:DateTime>
            </nc:StartDate>
            <nc:EndDate>
              <nc:DateTime>2017-11-27T20:53:28Z</nc:DateTime>
            </nc:EndDate>
          </mof:EventValidityDateTimeRange>
          <mof:EventMessageDateTime>
            <nc:DateTime>2017-11-27T20:23:28Z</nc:DateTime>
          </mof:EventMessageDateTime>
          <emlc:ResourceDetail>
            <emlc:ResourceStatus>
              <emlc:ResourcePrimaryStatus>AVAILABLE</emlc:ResourcePrimaryStatus>
              <emlc:ResourceSecondaryTextStatus SourceID="PI.LOC">
                <emlc:Description>OD</emlc:Description>
              </emlc:ResourceSecondaryTextStatus>
            </emlc:ResourceStatus>
            <emlc:ResourceControllingOrganization>
              <nc:OrganizationIdentification>
                <nc:IdentificationID>PI.LOC</nc:IdentificationID>
              </nc:OrganizationIdentification>
              <emlc:ResourceIdentifier>
                <nc:IdentificationID>PI.LOC.PI999</nc:IdentificationID>
              </emlc:ResourceIdentifier>
            </emlc:ResourceControllingOrganization>
          </emlc:ResourceDetail>
        </emlc:Event>
      </embeddedXMLContent>
    </xmlContent>
  </contentObject>
</EDXLDistribution>

```

SUMMARY

This how-to guide is intended to allow potential users to configure and setup a Raspberry Pi with the PiPoint software to provide location services to other systems identified in the Introduction system of this document.

If your agency finds this NGFR case study useful for improving your deployable communications capabilities, the NGFR Apex program would greatly appreciate your feedback. Please contact the NGFR team with stories from the field, questions or comments by emailing NGFR@hq.dhs.gov.

REFERENCES & RECOMMENDED READING

DHS Science & Technology Directorate

Next Generation First Responder Apex Program (<https://dhs.gov/ngfr>)

This website provides NGFR Apex program descriptions, updates and knowledge products.

First Responder Publications (<https://www.dhs.gov/science-and-technology/frg-publications>)

Next-Generation First Responder (NGFR)

<https://www.dhs.gov/publication/st-frg-ngfr-integration-handbook-version-20>

Internet of Things (IoT)

https://www.dhs.gov/sites/default/files/publications/IMIS_IoT_Pilot_Architecture_Engineering_Report-Final-v2-508.pdf

https://www.dhs.gov/sites/default/files/publications/IoT_Extension_Report_OGC_final-508.pdf

<https://www.dhs.gov/sites/default/files/publications/Incident-Management-Information-Sharing-Internet-of-Things-Protocol-Mapping-Engineering-Report-508.pdf>

<https://www.dhs.gov/sites/default/files/publications/Incident-Management-Information-Sharing-Profile-Recommendations-for-OGC-Web-Services-Engineering-Report-508.pdf>