Content:

Writing an OS Installation Image to Flash Media

Flash Media includes:

- USB drives
- Compact Flash cards
- SD cards
- Micro SD cards

Download an Installation Image

Download an installation image file from the appropriate location to a workstation from which the file will be written.

Make sure to choose the version of the image for the correct architecture. Most operating systems use either **amd64** or **x86_64** as the designation to represent the 64-bit architecture used by most modern processors. The 32-bit predecessor of this is usually designated i386.

Verify the Downloaded Image

Verify that the download succeeded by verifying the hash of the image file. The download site will likely have a file with *sha256* or *md5* in the filename. Many times this is in the same HTTP or FTP directory from which the original image was retrieved. The hash of the downloaded file may be calculated and that value compared against the hash that the project specified.

Hash calculation programs vary by operating system, some common examples include:

- Windows: The HashTab application.
- Linux and Cygwin (Windows): `sha256sum` or `md5sum` command line utilities
- FreeBSD and Mac OSX: `sha256` or `md5` command line utilities

Decompress the Downloaded Image

If the downloaded image was compressed , before proceeding to the next steps, decompress it using an appropriate utility such as 7-Zip, `gunzip` or `bunzip2`.

The most common indication that a file is compressed is its extension. If the filename ends in `.gz`,`.bz2`, `.zip`, or similar then it is likely compressed.

Connect the Flash Drive to the Workstation

Start by connecting the drive to the workstation where the downloaded image resides.

Locate the device name that the system designates for the drive. Here are some pointers for what the device name may look like on different platforms:

- Linux: */dev/sdX* where *X* is a lowercase letter. Look for messages about the drive attaching in the system log files or by running `dmesg`.
- FreeBSD: */dev/daX* where *X* is a decimal digit. Look for messages about the drive attaching in the system log files or by running `dmesg`.
- Windows: the drive will be named after a single uppercase letter, e.g. *D*. Use Explorer or examine the system control panel and look at the available disks for one matching the drive.

- On Mac OS X: */dev/diskX* where X is a decimal digit. Rrun `diskutil list` from a command prompt or use the GUI tool **Disk Utility**.

  Also make sure the located name refers device itself rather than a partition on the device. For example, */dev/sdb1* on Linux is the first partition on the disk, so it would be writing to a partition on the device and the drive may not end up being bootable. In that case use */dev/sdb* instead so the entire disk is written

Write the Image

Now it is time to write the image to the flash drive. The exact procedure varies by Operating System.

Linux, FreeBSD, Mac OS X

On Linux, FreeBSD and Mac OSX, write the image to the drive using the *dd* command. It takes this general form:

```
dd if=image_file_name of=usb_disk_device_name
```

Some examples for different platforms:

- Linux:
- `sudo dd if=file1.img of=/dev/sdb bs=4M`
- FreeBSD:
- `sudo dd if=file1.img of=/dev/da1 bs=4m`
- Mac OSX:
- `sudo dd if=file1.img of=/dev/rdisk3 bs=4m`

The `bs=X` is optional and tells `dd` to perform reads and writes on 4 MB blocks of data at a time. The default block size used by `dd` is 512 bytes. Specifying a larger block size can significantly increase the writing speed which will result in faster image writing.

Windows

In order to write an image to a drive from a Windows workstation, use a GUI tool such as Win32 Disk Imager or Rufus. It is also possible to use the same Linux dd command listed above within Cygwin if the Cygwin command prompt is launched as an Admin user.

Connecting to RCC Series Console Port

Simple Configuration

Below are the simple instructions for connecting to the console port with Microsoft Windows. If these steps do not work for you or if you're an operating system other than Windows, then please skip forward to Advanced Configuration.

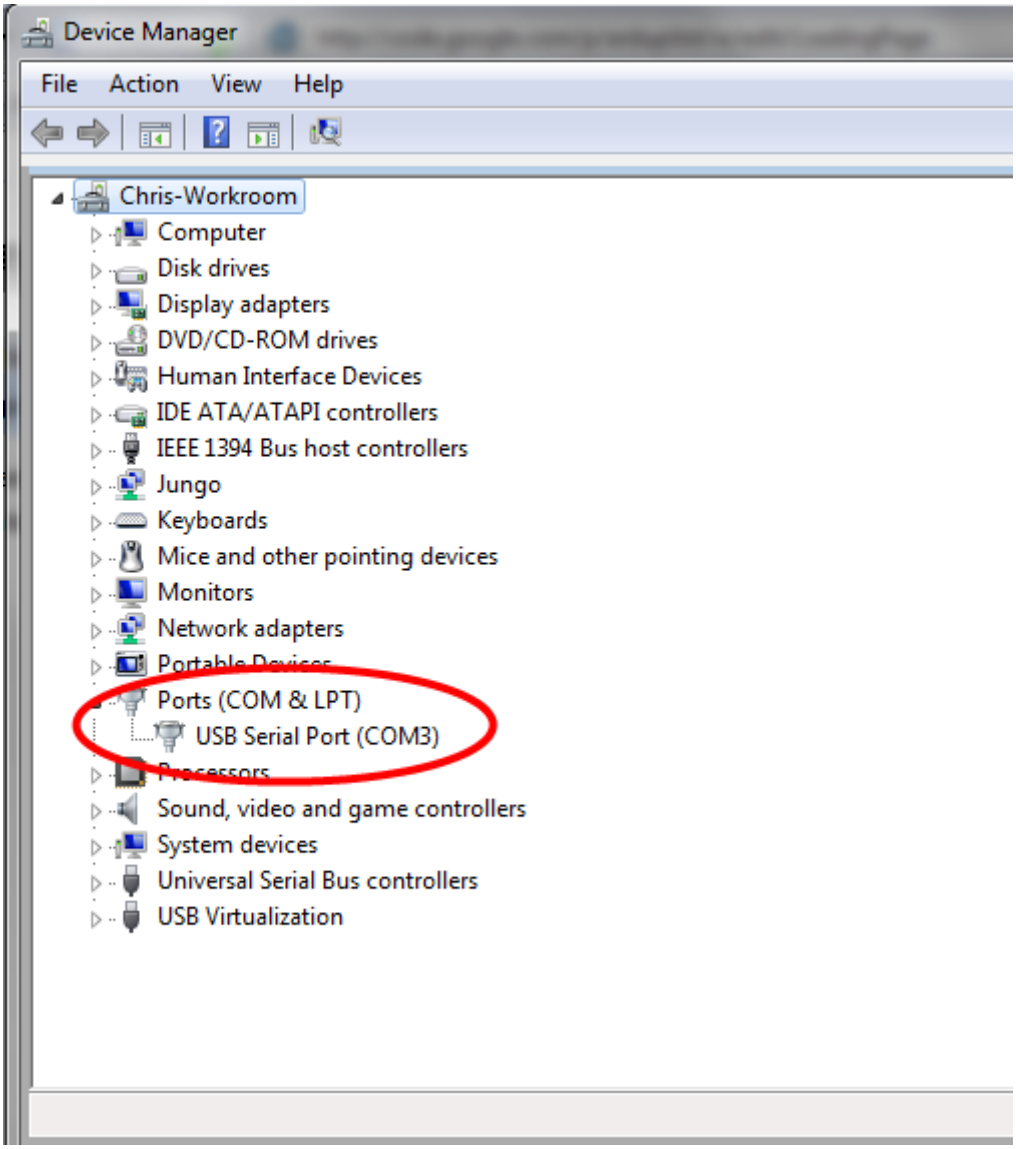
Serial Terminal Emulation Client

A serial terminal emulation program is required to access the pfSense appliance console through the serial interface. Microsoft Windows no longer includes HyperTerminal in Versions 7 and up. PuTTY is free and can be downloaded from:

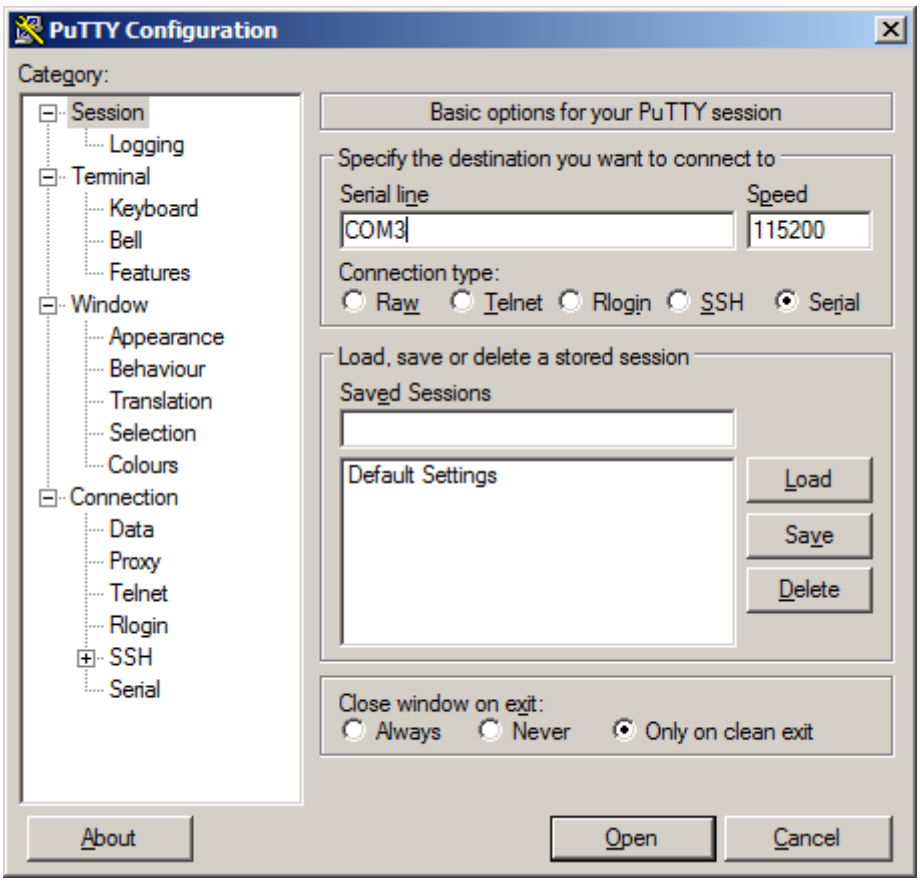
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Configuring Serial Terminal Emulator

PuTTY must be configured to communicate with the pfSense appliance. In order to do so, you must first know what COM Port your computer has assigned to your serial port. Even if you assigned your serial port to COM1 in the BIOS, Windows may remap it to a different COM Port. To determine this, you must open Windows Device Manager and view the COM port assignment.



Open PuTTY and locate the Session display as shown. Set the COM Port to that which is displayed in Windows Device Manager and the Speed to 115200.



Match the COM Port with what was reported in Windows Device Manager. We will use COM3 for this example. The pfSense appliance serial port speed is 115200 bits per second. The speed of the BIOS and the speed of the console must match so change the speed in PuTTy to 115200bps.

Select `Serial` as shown and configure the COM Port and Serial Speed as displayed. Select `Open` and the conolse screen will be displayed.

Advanced Configuration

The RCC Series uses a Silicon Labs CP210x USB-to-UART bridge to provide access to the serial port that acts as a system console. This is exposed via a mini-USB port on the front of the case. There are several steps required to access the system console via this port.

Install the Driver

Install an appropriate CP210x USB to UART Bridge VCP (virtual COM port) driver on the workstation used to connect with the RCC Series unit if needed. There are drivers available for Windows, Mac OS X, and Linux available in the Download Software section of the Silicon Labs Website.

Attention. Do not download the **SDK**, only download the driver

Recent versions of FreeBSD and many Linux distributions include this driver and will not require manual installation

Loading the Linux Driver

If the device does not appear automatically, the CP210x driver module may need to be loaded manually, especially if the version of Linux being run is not recent. If the driver was provided with the Linux distribution,

run `modprobe cp210x` as root or using `sudo`. If it had to be built manually, run `insmod ./cp210x.ko` assuming the module is in the current directory.

Connect a USB Cable

Next, locate an appropriate USB cable. The type of cable required for the serial console has a mini USB (Mini-B) connector on one end and a regular USB (Type A) plug on the other end. These cables are commonly used with smaller USB peripherals such as GPS units, cameras, and so on. They can be found in the pfSense Store and other retailers.

Attach the USB cable between a workstation and the RCC Series unit. Gently push the Mini-B plug end into the console port on the RCC Series unit and connect the USB type A plug into an available USB port on the workstation.

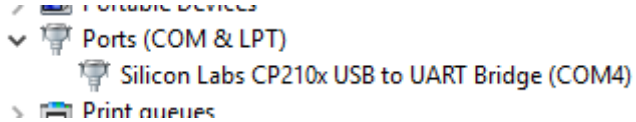
Be certain to gently push in the Mini-B connector on the RCC Series unit side completely. With most cables there will be a tangible “click”, “snap”, or similar indication when the cable is fully engaged

Locate the Console Port Device

The appropriate device to attach the terminal program to each platform varies by platform and must be located before attempting to connect to the console.

Windows

To locate the device name on Windows, open **Device Manager** and expand the section for **Ports (COM & LPT)**. Look for an entry with a title such as **Silicon Labs CP210x USB to UART Bridge**. If there is a label in the name that contains “COMX” where X is a decimal digit (e.g. COM1), that value is what would be used as the port in the terminal program.



Mac OS OX

The device associated with the RCC Series console is likely to show up as */dev/cu.SLAB_USBtoUART*.

Linux

The device associated with the RCC Series console is likely to show up as */dev/ttyUSBX* (where *X* is a decimal digit) on a linux system. Look for messages about the device attaching in the system log files or by running `dmesg`.

If the device does not appear in */dev/*, see the note above in the driver section about manually loading the Linux driver and then try again

FreeBSD

The device associated with the RCC Series console is likely to show up as */dev/cuaUX* (where *X* is a decimal digit). Look for messages about the device attaching in the system log files or by running `dmesg`.

Launch a Terminal Program

Use a terminal program to connect to the system console port. PuTTY is a popular terminal program that is available on various operating systems. Some other choices of terminal programs:

- Linux: screen, PuTTY, minicom, dterm
- Mac OS X: screen, ZTerm, cu
- Windows: PuTTY, SecureCRT, **Do not use Hyperterminal**
- FreeBSD: screen, cu

The settings to use within the terminal program are:

Speed: 115200 baud

Data bits: 8

Parity: none

Stop bits: 1

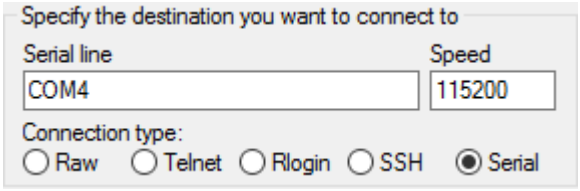
Flow Control: Off or XON/OFF. Hardware flow control (RTS/CTS) **must** be disabled.

Client-Specific Examples

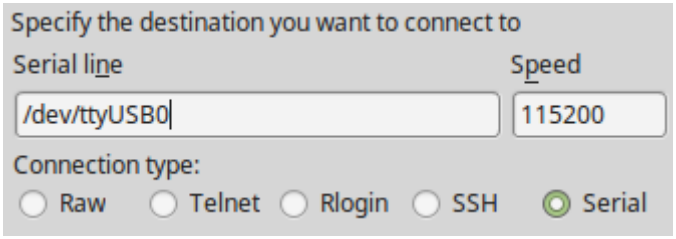
PuTTY

Launch PuTTY and configure it for a **Serial** type connection with a speed of **115200** using the port name located previously.

- Windows Example:



- Linux Example:



PuTTY generally handles most cases OK but can have issues with line drawing characters on certain platforms.

These settings seem to work best (tested on Windows):

Window: Columns x Rows = *80x24*

Window > Appearance:

Font = *Courier New 10pt* or *Consolas 10pt*

Window > Translation:

Remote Character Set = *Use font encoding* or *UTF-8*

Window > Translation:

Handling of line drawing characters = *Use font in both ANSI and OEM modes* or *Use Unicode line drawing code points*

Window > Colours:

Indicate bolded text by changing = The colour

GNU screen

In many cases *screen* may be invoked simply by using the proper command line:

- Mac OS X:
- `sudo screen /dev/cu.SLAB_USBtoUART 115200`
- Linux:
- `sudo screen /dev/ttyUSB0 115200`
- FreeBSD:
- `sudo screen /dev/cuaU0 115200`

If portions of the text are unreadable but appear to be properly formatted, the most likely culprit is a character encoding mismatch in the terminal. For example, on OS X this is commonly required:

```
sudo screen -U /dev/cu.SLAB_USBtoUART 115200
```

Adding the `-U` parameter to the *screen* command line arguments forces it to use UTF-8 for character encoding.

Troubleshooting

No Serial Output

If there is no output at all, check the following items:

- Ensure the cable is correctly attached and fully inserted
- Ensure the terminal program is using the correct port
- Ensure the terminal program is configured for the correct speed. The default BIOS speed on RCC Series units is 115200, and many other modern operating systems use that speed as well. Some older operating systems or custom configurations may use slower speeds such as 9600 or 38400.
- Ensure the operating system is configured for the proper console (e.g. `ttyS1` in Linux). Consult the various operating install guides on this site for further information.

Garbled Serial Output

If the serial output apears to be garbled, binary, or random characters check the following items:

- Ensure the terminal program is configured for the correct speed. (See “No Serial Output” above)
- Ensure the terminal program is configured for the proper character encoding, such as UTF-8 or Latin-1, depending on the operating system. (See the previous entry under “GNU screen”)

Serial Output Stops After the BIOS

If serial output is shown for the BIOS but stops afterward, check the following items:

- Ensure the terminal program is configured for the correct speed for the installed operating system. (See “No Serial Output” above)
- Ensure the installed operating system is configured to activate the serial console.
- Ensure the installed operating system is configured for the proper console (e.g. `ttyS1` in Linux). Consult the various operating install guides on this site for further information.
- If booting from a USB flash drive, ensure that the drive was written correctly and contains a bootable operating system image.