**Basic Installation**

Download Debian Wheezy image (2012-08-08-wheezy-armel.zip) from Raspberry Pi homepage.

Unzip image and write it to SD card.

# dd bs=4M if=/home/seelmann/Development/RPi/2012-08-08-wheezy-armel.img of=/dev/mmcblk0

Mount partitions.

Use maximum memory for applications as the Pi is operated in headless mode and no video output is required.

# cp arm240\_start.elf start.elf

Create link to start SSH server:

# cd etc/rc2.d

# ln -s ../init.d/ssh S03ssh

Set a static IP address:

# cat /etc/network/interfaces

auto lo

iface lo inet loopback

auto eth0

iface eth0 inet static

address 192.168.2.2

netmask 255.255.255.0

gateway 192.168.2.1

# cat /etc/resolv.conf

nameserver 192.168.2.1

Boot the Pi.

Connect via SSh.

$ ssh pi@192.168.2.2

Run basic Pi configuration:

$ sudo raspi-config

* Change password for 'pi' user
* Set timezone
* Enable SSH server
* Disable desktop start on boot

Config java: $**sudo** update-alternatives --config java

Run *Command Prompt* (found under All Programs/Accessories in the Start menu).  Type

**C:\> cd \mywork**

This makes C:\mywork the current directory.

**C:\mywork> dir**

This displays the directory contents.  You should see HelloWorld.java among the files.

**C:\mywork> set path=%path%;C:\Program Files\Java\jdk1.5.0\_09\bin**

This tells the system where to find JDK programs.

**C:\mywork> javac HelloWorld.java**

This runs javac.exe, the compiler.  You should see nothing but the next system prompt...

**C:\mywork> dir**

*javac* has created the HelloWorld.class file.  You should see HelloWorld.java andHelloWorld.class among the files.

**C:\mywork> java HelloWorld**

This runs the Java interpreter.  You should see the program output:

Hello, World!

If the system cannot find *javac*, check the set path command.  If *javac* runs but you get errors, check your Java text.  If the program compiles but you get an exception, check the spelling and capitalization in the file name and the class name and the java HelloWorldcommand.  Java is case-sensitive!

Delete unused software packages, in reverse order of /var/log/apt/history.log, including automatically installed packages:

# apt-get purge --auto-remove scratch

# apt-get purge --auto-remove debian-reference-en dillo idle3 python3-tk idle python-pygame python-tk

# apt-get purge --auto-remove lightdm gnome-themes-standard gnome-icon-theme raspberrypi-artwork

# apt-get purge --auto-remove gvfs-backends gvfs-fuse desktop-base lxpolkit netsurf-gtk zenity xdg-utilse

# apt-get purge --auto-remove mupdf gtk2-engines alsa-utils lxde lxtask menu-xdg gksu

# apt-get purge --auto-remove midori xserver-xorg xinit xserver-xorg-video-fbdev

# apt-get purge --auto-remove libraspberrypi-dev libraspberrypi-doc

# apt-get purge --auto-remove dbus-x11 libx11-6 libx11-data libx11-xcb1 x11-common x11-utils

# apt-get purge --auto-remove lxde-icon-theme gconf-service gconf2-common

Upgrade

# apt-get update

# apt-get dist-upgrade

**Print Server**

# apt-get install cups

Allow remote administration:

# cupsctl --share-printers --remote-admin --remote-printers

Add pi user to lpadmin group:

# usermod -a -G lpadmin pi

Use CUPS web UI:

https://192.168.2.2:631/admin/

**Scan Server**

# apt-get install sane-utils libsane

Test if scanner is supported:

# sane-find-scanner

# scanimage -L

Allow remote access

# cat /etc/sane.d/saned.conf

192.168.2.0/24

Activate saned:

# cat /etc/default/saned

RUN=yes

RUN\_AS\_USER=saned

# /etc/init.d/saned start

On the client add the IP of the scan server to /etc/sane.d/net.conf:

# cat /etc/sane.d/net.conf

192.168.2.2

**Chat Server**

# apt-get install prosody

# prosodyctl adduser foo@localhost