

Build Environment - Linux

Linux - NOTE: In all cases you MUST install a 32-bit OS (Either CentOS, Fedora, or Ubuntu)

The Linux build environment documented on this page is based on a basic [Fedora Core 15](#) installation CD. There are CentOS 6- and Ubuntu-specific notes below. It should also be possible to use a previously existing Linux installation, assuming that you have all of the required packages described below. The following section describes how to set up a machine from scratch.

Preparing a System for Build

Install Fedora Core 15 i386 CD or CentOS 6.x or Ubuntu 12.04 LTS (64 bit not supported. Note specific platform notes below, and at bottom.)

1. !* Choose and make note of the root password
 - Create your user (remember the password)
2. Update the software on the system
 - System->Administration->Update System
 - If you get the following error "Transaction error could not add package update for <distro info>". The work around is typing the following on the command line:
\$su -
\$yum update
3. Convert your user to a "sudoer"
 - Get a terminal window (Applications->System Tools->Terminal)
 - Type 'su -' (no quotes)
 - Enter root password
 - Fedora, CentOS
 - visudo
 - add the following line toward the end of the file (assuming your user name was steve)
%steve ALL=(ALL) ALL
 - Ubuntu
 - Add user to "sudo" group in /etc/group in 12.04 LTS, or "admin" group in earlier versions, e.g.
sudo:x:27:steve
admin:x:25:steve
4. Install all the required Linux packages (CentOS, Fedora. See bottom for Ubuntu).
 - cut and paste these to the terminal window:
 - Fedora only
sudo rpm -ivh <http://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-stable.noarch.rpm>
 - CentOS 6 only
 - sudo rpm -ivh <http://mirrors.servercentral.net/fedora/epel/6/i386/epel-release-6-8.noarch.rpm>
 - sudo rpm -ivh <http://download1.rpmfusion.org/free/el/updates/6/i386/rpmfusion-free-release-6-1.noarch.rpm>
 - sudo yum clean all (remove "RPMDB altered outside of yum" warning, if necessary)
 - Install necessary packages

```
sudo yum install gcc gcc-c++ make gettext patch bison flex liboil liboil-devel zlib zlib-devel  
libxml2 libxml2-devel doxygen libX11-devel libXext libXext-devel mesa-libGLU-devel atk atk-devel  
pango pango-devel cairo cairo-devel libjpeg libjpeg-devel libtiff libtiff-devel man-pages perl-Net-  
Telnet perl-Digest-SHA1 svn vlc mozilla-vc
```

5. Check the RI Code out to populate your machine
 - Start in your home directory ('cd')
 - mkdir CableLabsRI
 - cd CableLabsRI
 - svn co <https://community.cablelabs.com/svn/OCAPRI/trunk>
 - Enter your credentials when prompted, then wait as it checks out the trunk. You now have the majority of the tree for Linux on your machine under /home/<user>/CableLabsRI/trunk
6. Ensure SELinux is disabled (CentOS, Fedora. Ubuntu does not install and enable SELinux by default.)
 - Activate menu by selecting System->Administration->SELinux Management
 - Enter root password
 - In Status Tab, set System Default Enforcing Mode to Disable
 - For Fedora 15 and CentOS:
 - sudo vi /etc/sysconfig/selinux
 - change SELINUX=enforcing to SELINUX=disabled
 - :wq (save and quit).
 - You have to reboot the system for this to take effect.
7. Install the Sun 1.6 Java Development Kit (Java SE JDK 6 Update XX)
 - Note: Installation of Sun's JDK is not necessary - the Open JDK has already been installed (and works). However, the Open JDK developer tools must be installed (i.e. the full JDK, not just the JRE):
 - For CentOS:

```
yum install java-1.6.0-openjdk java-1.6.0-openjdk-devel
```

For Ubuntu:
See full apt-get command at bottom.

- **For Sun JDK:**
 - Go to <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
 - Select latest JDK 6 Download
 - Select Platform: Linux Language: Multi-language, select checkbox accepting the License agreement
 - Select jdk_6uxx-linux-rpm.bin, and "Save File" when prompted
 - in ~/Download, do 'chmod +x jdk-6uxx-linux-i586-rpm.bin' (no quotes)
 - sudo ./jdk-6uxx-linux-i586-rpm.bin
 - Accept the license, and watch the jdk get installed
 - Ignore the registration
8. Get ant 1.8 or later
- ANT is already installed. However, DO NOT SET ANT_HOME
 - Download <http://ant.apache.org/bindownload.cgi>
 - in home directory do 'gtar -xvf Download/apache-ant-1.8.4-bin.tar.gz'
9. Edit your environment to get ready to build
- Edit .bashrc in your home directory
 - Add the following lines to the end, replacing "steve" with your user name

```
#####
# Env variables required for building the OCAP Stack and JVM
#
# Notice the use of "forward slashes" in all paths.  This is
# required for the Cygwin build system.
#####

# The root of the stack source tree
export OCAPROOT=~steve/CableLabsRI/trunk/ri/RI_Stack

# Identifies the host development platform.  See $OCAPROOT/hostconfig
export OCAPHOST=Linux
#export OCAPHOST=wxWidgets

# Stack build target
export OCAPTC=CableLabs/simulator/Linux/debug

# Ant and Java Home directories
export JAVA_HOME=/usr/java/jdk1.6.0_13
export ANT_HOME=~/apache-ant-1.8.4

# Add Java, Ant, and Doxygen bin directories to the path.  Notice the use of
# Cygwin drive designations instead of Windows
export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$PATH:~/atelite/bin:

#####
# Env variables required for building the RI Platform
#
# Notice the use of "forward slashes" in all paths.  This is
# required for the Cygwin build system.
#####

# The directory root of the RI platform source tree
export PLATFORMROOT=~steve/CableLabsRI/trunk/ri/RI_Platform
# Identifies the host development platform.  See $PLATFORMROOT/hostconfig
export PLATFORMHOST=Linux
# Platform build target
export PLATFORMTC=Linux/debug

# For the Manufacturers Enviornement
export MFGROOT=c:/CableLabsRI/Subversion/MfgEnv
export MFGTC=Win32/debug

export RICOMMONROOT=~steve/CableLabsRI/trunk/common
export HE4RIROOT=~steve/CableLabsRI/trunk/emu/HeDri/Emulator
```

Building Everything for the First Time

1. Build the platform (the support code for the OCAP stack)
 - Start a new terminal session to pick up the environment changes you just made

- `cd $PLATFORMROOT`
 - `make clean purge build`
2. Build the OCAP stack (the Java VM and support libraries)
 - `cd $OCAPROOT`
 - `ant clean`
 - `ant purge` (this may fail due to a bug in the build script for now - ignore)
 - `ant build`
 - This should build the stack - Yahoo! Wait for complete with no errors
 - From this point forward, 'ant purge' should work, so you can do a super-clean build using 'ant clean purge build'

Subsequent Builds

1. Building the platform (assuming you did not change the support libraries)
 - From \$PLATFORMROOT do 'build source'

Required Linux Packages

You should first install all of the recommended additional packages that the Fedora system manager directs you to.

The following packages must be installed on your Linux system prior to building the RI Platform and Stack. You can download and install packages directly from Fedora. Click on **System > Administration > Add/Remove Software**. Sometimes you will need to look for older packages (e.g. Subversion 1.4). For those packages the [Fedora Package Database](#) can prove to be a valuable resource.

Package Name	Description	Known Working Version	Comments
gcc	GNU Compiler Collection	4.3.2-7	Installing this package will also install a lot of other packages such as binutils, glibc, etc
gcc-c++	C++ support for GCC	4.3.2-7	Installing this package will also install libstdc++-devel
make	GNU Build Tool	1.3.81-14	
gettext	Multilingual message library	0.17-8	
patch	Utility for modifying/upgrading files	2.5.4-35	
bison	Parser generator	2.3-5	
flex	Tool for creating scanners	2.5.35-2	
liboil	Library of Optimized Inner Loops	0.3.14-1	May already be installed
liboil-devel	Development files for liboil	0.3.14-1	
zlib	ZLib compression/decompression library	1.2.3-18	May already be installed
zlib-devel	Headers and libs for zlib development	1.2.3-18	
libxml2	Library providing XML and HTML support	2.7.3-1	May already be installed
libxml2-devel	Libraries for developing XML/HTML apps	2.7.3-1	
doxygen	Documentation generator for C/C++	1.5.7.1-1	
libX11-devel	X11 development package	1.1.4-6	Installing this package will install several other X11 development packages (including Mesa libGL)
libXext	X11 libXext runtime library	1.0.4-1	May already be installed
libXext-devel	X11 libXext development package	1.0.4-1	
mesa-libGLU-devel	Mesa libGLU development package	7.2-0.15	
atk	Accessibility support	1.24.0-1	May already be installed
atk-devel	ATK development libs	1.24.0-1	Installing this package will install several other packages
pango	Layout for internationalized text	1.22.3-1	
pango-devel	Development libs for pango	1.22.3-1	
cairo	Graphics library	1.8.0-1	
cairo-devel	Development libs for cairo	1.8.0-1	
libjpeg	JPEG libs	6b-43	
libjpeg-devel	JPEG development libs	6b-43	
libtiff	TIFF libs	3.8.2-11	

libtiff-devel	TIFF development libs	3.8.2-11	
man-pages	Lots of good man pages for Linux	3.09-2	Obviously, this is optional, but it installs man pages for glibc functions which are always helpful
perl-Net-Telnet	Telnet libraries for perl	3.03-7	Required for ATE Emulator
perl-Digest-SHA1	SHA1 libraries for perl	2.11.7	Required for ATE Emulator
pkg-config	Linux package configuration utility		
ant	Ant make tool	1.8.0	Required to build the OCAP stack
ant-jdepend	Ant support libraries for dependency checking	1.7.1	Required to build the OCAP stack
vlc	Media Streamer/Player	0.9.9-2	Used by the RI Platform to play MPEG-2 transport streams from file

Add your user group to sudoers file

*(Fedora/CentOS-only) In order to run ATE tests, the RI Platform must be able to connect to certain network ports that are only available to processes running as root. To facilitate this, we have modified the *runPlatform_linux.sh* startup script to always execute the RI process as root. On Linux systems, non-root users must add their user group to a list of groups that are allowed to gain root privileges. The list of these groups is found in the **/etc/sudoers** file. Edit this file and add your user group to one of the two sections

```
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
## user MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
# %wheel    ALL=(ALL)    ALL          ----->      Add your group using this
syntax if you want                                     to be required to enter a

password each time
## Same thing without a password
# %wheel    ALL=(ALL)    NOPASSWD: ALL
%myusergroup ALL=(ALL)    NOPASSWD: ALL      ----->      Add your group using this syntax if you DO
not                                                    want to be required to enter a password

each time
## Allows members of the users group to mount and unmount the
## cdrom as root
# %users    ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
```

Selecting a User Interface Target

On the Linux platform, there are currently two user interface targets:

- Native libX11 Windows
- wxWidgets Windows (default)

To build the native X11 Windows UI target, uncomment the line

```
UI_TARGET = Linux
```

in the *target/Linux/debug/defs.mk* file. To build for the cross-platform wxWidgets UI target, uncomment the line

```
UI_TARGET = wxWidgets
```

in the `target/Linux/debug/defs.mk` file. Make sure that the target you are NOT interested in is also commented out.

Ubuntu Specific Notes

Ubuntu 10.04.01 LTS, 10.10, the following command will install all prerequisite packages.

```
sudo apt-get install gcc make gettext patch bison flex liboil0.3 liboil0.3-dev libxml2 libxml2-dev doxygen
libX11-dev libxext6 libxext-dev libglu1-mesa-dev libatk1.0-0 libatk1.0-dev libpango1.0-common libpango1.0-dev
libcairo2 libcairo2-dev libtiff4 libtiff4-dev manpages libnet-telnet-perl libdigest-sha1-perl subversion ant1.8
libjdepend-java openjdk-6-jdk vlc
```

Ubuntu 11.04, 11.10 the following command will install all prerequisite packages.

```
sudo apt-get install gcc make gettext patch bison flex liboil0.3 liboil0.3-dev libxml2 libxml2-dev doxygen
libX11-dev libxext6 libxext-dev libglu1-mesa-dev libatk1.0-0 libatk1.0-dev libpango* libcairo2 libcairo2-dev
libtiff4 libtiff4-dev manpages libnet-telnet-perl libdigest-sha1-perl subversion ant libjdepend-java openjdk-6-
jdk vlc xsltproc
```

Ubuntu 12.04 LTS prerequisite packages

```
sudo apt-get install gcc make gettext patch bison flex liboil0.3 liboil0.3-dev libxml2 libxml2-dev doxygen
libX11-dev libxext6 libxext-dev libglu1-mesa-dev libatk1.0-0 libatk1.0-dev libpango* libcairo2 libcairo2-dev
libtiff4 libtiff4-dev manpages libnet-telnet-perl libdigest-sha-perl subversion ant libjdepend-java openjdk-6-
jdk vlc xsltproc
```

fix SHA1.pm renaming issue

```
sudo cpan Digest::SHA1
```

If you do a fresh install of 12.04 (as opposed to an upgrade from 11.x) and need build CTP test streams run:

```
sudo cpan HTTP::Request
sudo cpan LWP::UserAgent.pm
```

All Ubuntu

Add these lines to `~/.bashrc`:

```
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk
export PATH=$PATH:~/atelite/bin
```

```
export ANT_HOME=/usr/share/ant
```

OR

```
export ANT_HOME=~/apache-ant-1.8.4
```

If you have scripts or other that use `/bin/sh`, configure to use bash instead of dash:

```
sudo dpkg-reconfigure dash
```

...and answer **No**