

# **V2101 Series Linux User's Manual**

---

**Second Edition, November 2013**

[www.moxa.com/product](http://www.moxa.com/product)

**MOXA<sup>®</sup>**

© 2009 Moxa Inc. All rights reserved.

# V2101 Series Linux User's Manual

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

## Copyright Notice

© 2009 Moxa Inc. All rights reserved.

## Trademarks

The MOXA logo is a registered trademark of Moxa Inc.  
All other trademarks or registered marks in this manual belong to their respective manufacturers.

## Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document as is, without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

## Technical Support Contact Information

[www.moxa.com/support](http://www.moxa.com/support)

### **Moxa Americas**

Toll-free: 1-888-669-2872  
Tel: +1-714-528-6777  
Fax: +1-714-528-6778

### **Moxa Europe**

Tel: +49-89-3 70 03 99-0  
Fax: +49-89-3 70 03 99-99

### **Moxa India**

Tel: +91-80-4172-9088  
Fax: +91-80-4132-1045

### **Moxa China (Shanghai office)**

Toll-free: 800-820-5036  
Tel: +86-21-5258-9955  
Fax: +86-21-5258-5505

### **Moxa Asia-Pacific**

Tel: +886-2-8919-1230  
Fax: +886-2-8919-1231

# Table of Contents

<b>1. Introduction</b>	<b>1-1</b>
Overview	1-2
Software Specifications	1-2
Software Components	1-3
<b>2. Software Configuration</b>	<b>2-1</b>
Starting from a VGA Console	2-2
Connecting from a Telnet Console	2-2
Connecting from an SSH Console	2-3
Windows Users	2-3
Linux Users	2-4
Adjusting the System Time	2-4
Setting the Time Manually	2-4
NTP Client	2-4
Updating the Time Automatically	2-5
Enabling and Disabling Daemons	2-6
Setting the Run-Level	2-8
Cron—Daemon for Executing Scheduled Commands	2-9
Inserting a USB Storage Device into the Computer	2-10
VGA & LVDS Configuration	2-10
Checking the Linux Version	2-14
APT—Installing and Removing Packages	2-14
<b>3. Managing Communications</b>	<b>3-1</b>
Configuring Network Interfaces	3-2
Configuring a Persistent Network Interface Naming Order	3-2
Ethernet Interface Configuration	3-3
Static IP Address	3-3
Dynamic IP Address using DHCP	3-3
Adjusting IP Addresses with ifconfig	3-4
Point-to-Point Over Ethernet (PPPoE) Config	3-4
The Easy Way: pppoeconf	3-4
The Difficult Way (Manually)	3-6
Configuring a Point-to-Point Connection	3-7
Connecting to a PPP Server over a Hardwired Link	3-8
Checking the Connection	3-9
Setting up a Machine for Incoming PPP Connections	3-10
Serial Port Operation Mode	3-11
Telnet/FTP/TFTP Server	3-11
Enabling a Telnet, FTP, or TFTP Server	3-11
Disabling a Telnet/FTP/TFTP Server	3-12
DNS Utilities	3-12
Configuring the OS Hostname	3-12
Configuring the DNS Resolver	3-12
Configuring the Name Service Switcher	3-13
Apache Web Server	3-13
Default Homepage	3-14
Configuring the Common Gateway Interface (CGI)	3-14
Saving Web Pages to a USB Storage Device	3-15
Netfilter/iptables	3-17
IP Tables and IP Chains	3-17
Understanding Rule Chains	3-18
Understanding Basic Traffic Flows	3-19
Connection Tracking	3-21
Building the Firewall	3-21
Policies: Setting Default Firewall Behavior	3-21
Viewing and Manipulating Rulesets	3-23
Writing Rulechains	3-24
Setting Up NAT (Network Address Translation)	3-26
Saving the Firewall	3-27
Setting up a Networked File System: NFS	3-27
Setting Up a VPN	3-28
SNMP (Simple Network Management Protocol)	3-33
<b>4. Programmer's Guide</b>	<b>4-1</b>
Device API	4-2
RTC (Real Time Clock)	4-2
UART	4-2
Digital I/O	4-5
<b>5. Built-In Linux OS Recovery</b>	<b>5-1</b>

Recovery Environment .....	5-2
Recovery Procedure .....	5-2
Step 1: Format an Empty USB Disk.....	5-2
Step 2: Create a Linux Bootable USB Disk.....	5-2
Step 3: Set up the BIOS to Boot from a USB Disk.....	5-3
Step 4: Recover the Linux system from a USB Disk.....	5-5
Step 5: Reset the BIOS to boot from DOM or CompactFlash.....	5-7
<b>A. Sample Scripts &amp; Firewalls.....</b>	<b>A-1</b>
A Sample Initialization Script .....	A-2
A Sample Firewall .....	A-4

## Introduction

---

Thank you for purchasing the Moxa V2101 Series of x86 ready-to-run embedded computers. This manual introduces the software configuration and management of the V2101-LX, which runs the Linux operating system. For hardware installation, connector interfaces, setup, and upgrading the BIOS, please refer to the "V2101 Series Hardware User's Manual."

Linux is an open, scalable operating system that allows you to build a wide range of innovative, small footprint devices. Software written for desktop PCs can be easily ported to the embedded computer with a GNU cross compiler and a minimum of source code modifications. A typical Linux-based device is designed for a specific use, and is often not connected to other computers, or a number of such devices connect to a centralized, front-end host. Examples include enterprise tools such as industrial controllers, communications hubs, point-of-sale terminals, and display devices, which include HMIs, advertisement appliances, and interactive panels.

The following topics are covered in this chapter:

- ❑ **Overview**
- ❑ **Software Specifications**
- ❑ **Software Components**

# Overview

V2101 embedded computers are based on the Intel Atom Menlow XL x86 processor and feature two serial ports, dual Gigabit LAN ports, four USB 2.0 hosts, and an SD socket. The V2100 series offers both VGA and LVDS outputs, making it exceptionally well suited for industrial applications such as SCADA and factory automation.

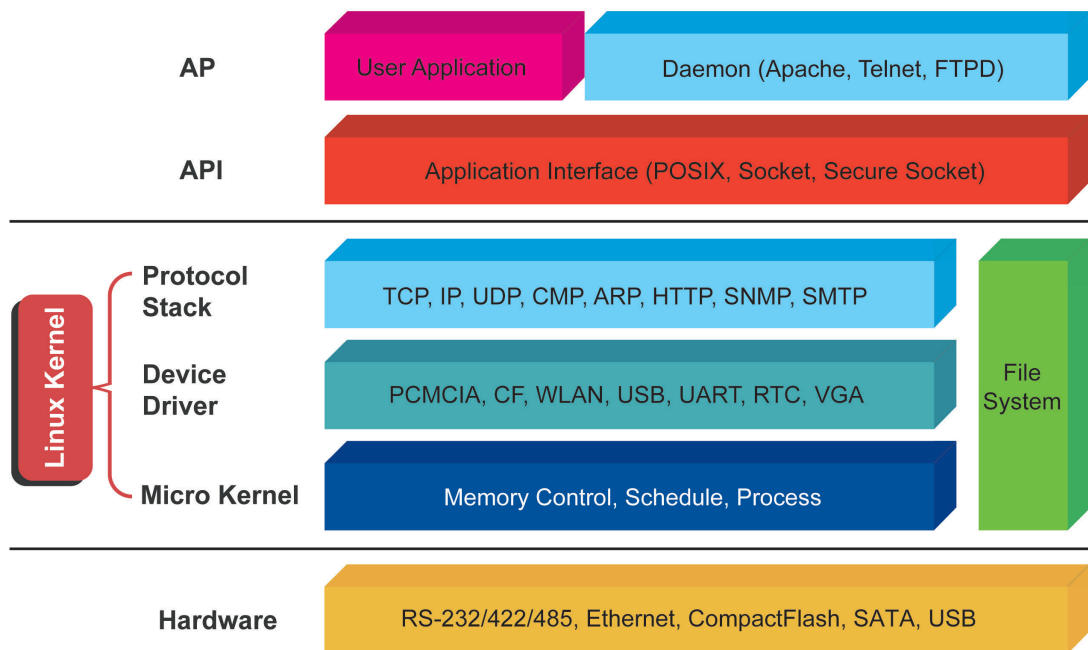
The V2101's two serial ports make it ideal for connecting a wide range of serial devices, and the dual 10/100/1000 Mbps Ethernet ports offer a reliable solution for network redundancy, which taken together promise continuous data communication and management operations. For added convenience, the V2101 has three DIs and three DOs for connecting digital input/output devices. In addition, the SD and USB ports provide V2101 computers with data buffering and storage expansion, which provide the necessary reliability for industrial applications.

Pre-installed with Linux, the V2101 Series provides programmers with a friendly environment for developing sophisticated, bug-free application software at a lower cost.

All V2101 models support a wide operating temperature range of -40 to 85°C for use in harsh industrial environments.

# Software Specifications

The Linux operating system pre-installed on the V2101 embedded computer is the **Debian Lenny 5.0** distribution. The Debian project is a worldwide group of volunteers who endeavor to produce an operating system distribution that composed entirely of free software. The Debian GNU/Linux follows the standard Linux architecture, making it easy to use programs that meet the POSIX standard. Program porting can be done with the GNU Tool Chain provided by Moxa. In addition to Standard POSIX APIs, device drivers for Moxa UART and other special peripherals are also included. An example software architecture is shown below:



## ATTENTION

Refer to <http://www.debian.org/> and <http://www.gnu.org/> for information and documentation of the Debian GNU/Linux and free software concept.

**ATTENTION**

The above software architecture is only an example. Different models or different build revisions of the Linux operating system may include components not shown in the above graphic.

## Software Components

The Debian 5.0/Lenny Linux system installed on this computer includes the following software components:

acpi-support-base	0.109-11	scripts for handling base ACPI events such as the power button
acpid	1.0.8-1lenny1	Utilities for using ACPI power management
adduser	3.110	add and remove users and groups
alacarte	0.11.5-1	easy GNOME menu editing tool
alsa-base	1.0.17.dfsg-4	ALSA driver configuration files
alsa-utils	1.0.16-2	ALSA utilities
apache2	2.2.9-10+lenny6	Apache HTTP Server metapackage
apache2-doc	2.2.9-10+lenny2	Apache HTTP Server documentation
apache2-mpm-prefork	2.2.9-10+lenny6	Apache HTTP Server - traditional non-threaded model
apache2-utils	2.2.9-10+lenny4	utility programs for web servers
apache2.2-common	2.2.9-10+lenny6	Apache HTTP Server common files
app-install-data	2008.11.27	Application Installer Data Files
apt	0.7.20.2+lenny1	Advanced front-end for dpkg
apt-file	2.1.5	APT package searching utility -- command-line interface
apt-utils	0.7.20.2+lenny1	APT utility programs
aptitude	0.4.11.11-1~lenny1	terminal-based package manager
arj	3.10.22-6	archiver for .arj files
aspell	0.60.6-1	GNU Aspell spell-checker
aspell-en	6.0-0-5.1	English dictionary for GNU Aspell
audacity	1.3.5-2+lenny1	A fast, cross-platform audio editor
autoconf	2.61-8	automatic configure script builder
autoconf2.13	2.13-59	automatic configure script builder (obsolete version)
automake	1:1.10.1-3	A tool for generating GNU Standards-compliant Makefiles
automake1.4	1:1.4-p6-13	A tool for generating GNU Standards-compliant Makefiles
autotools-dev	20080123.1	Update infrastructure for config.{guess,sub} files
base-files	5lenny4	Debian base system miscellaneous files
base-passwd	3.5.20	Debian base system master password and group files
bash	3.2-4	The GNU Bourne Again SHell
binutils	2.18.1~cvs20080103-7	The GNU assembler, linker and binary utilities
bridge-utils	1.4-5	Utilities for configuring the Linux Ethernet bridge
bsdmainutils	6.1.10	collection of more utilities from FreeBSD
bsdutils	1:2.13.1.1-1	Basic utilities from 4.BSD-Lite
busybox	1:1.10.2-2	Tiny utilities for small and embedded systems
bzip2	1.0.5-1	high-quality block-sorting file compressor - utilities

capplets-data	1:2.22.2.1-2	configuration applets for GNOME 2 - data files
cdrdao	1:1.2.2-16	records CDs in Disk-At-Once (DAO) mode
console-common	0.7.80	basic infrastructure for text console configuration
console-data	2:1.07-11	keymaps, fonts, charset maps, fallback tables for console-tool
console-tools	1:0.2.3dbs-65.1	Linux console and font utilities
coreutils	6.10-6	The GNU core utilities
cpio	2.9-13	GNU cpio -- a program to manage archives of files
cpp	4:4.3.2-2	The GNU C preprocessor (cpp)
cpp-4.3	4.3.2-1.1	The GNU C preprocessor
cron	3.0pl1-105	management of regular background processing
dbus	1.2.1-5+lenny1	simple interprocess messaging system
dbus-x11	1.2.1-5+lenny1	simple interprocess messaging system (X11 deps)
debconf	1.5.24	Debian configuration management system
debconf-i18n	1.5.24	full internationalization support for debconf
debian-archive-keyring	2009.01.31	GnuPG archive keys of the Debian archive
debianutils	2.30	Miscellaneous utilities specific to Debian
deborphan	1.7.27	program that can find unused packages, e.g. libraries
Defoma framework	0.11.10-0.2	Debian Font Manager -- automatic font configuration
deskbar-applet	2.22.3.1-1	universal search and navigation bar for GNOME
desktop-base	5.0.3	common files for the Debian Desktop
desktop-file-utils	0.15-1	Utilities for .desktop files
dhcp3-client	3.1.1-6+lenny3	DHCP client
dhcp3-common	3.1.1-6+lenny3	common files used by all the dhcp3* packages
dialog	1.1-20080316-1	Displays user-friendly dialog boxes from shell scripts
dictionaries-common	0.98.12	Common utilities for spelling dictionary tools
diff	2.8.1-12	File comparison utilities
dmidecode	2.9-1	Dump Desktop Management Interface data
doc-base	0.8.20	utilities to manage online documentation
docbook-xml	4.5-6	standard XML documentation system, for software and systems
dpkg	1.14.25	Debian package management system
dvd+rw-tools	7.1-3	DVD+-RW/R tools
e2fslibs	1.41.3-1	ext2 filesystem libraries
e2fsprogs	1.41.3-1	ext2/ext3/ext4 file system utilities
ed	0.7-3	The classic unix line editor
eject	2.1.5+deb1-4	ejects CDs and operates CD-Changers under Linux
eog	2.22.3-2	Eye of GNOME graphics viewer program
esound-clients	0.2.36-3	Enlightened Sound Daemon - clients
esound-common	0.2.36-3	Enlightened Sound Daemon - Common files
ethtool	6+20080913-1	display or change Ethernet device settings
evolution-data-server	2.22.3-1.1+lenny2	evolution database backend server
evolution-data-server-common	2.22.3-1.1+lenny2	architecture independent files for Evolution Data Server
fam	2.7.0-13.3	File Alteration Monitor
file	4.26-1	Determines file type using "magic" numbers
findutils	4.4.0-2	utilities for finding files--find, xargs
fontconfig	2.6.0-3	generic font configuration library - support binaries
fontconfig-config	2.6.0-3	generic font configuration library - configuration
ftp	0.17-18	The FTP client



ftpd	0.17-29	File Transfer Protocol (FTP) server
g++	4:4.3.2-2	The GNU C++ compiler
g++-4.3	4.3.2-1.1	The GNU C++ compiler
gamin	0.1.9-2	File and directory monitoring system
gcc	4:4.3.2-2	The GNU C compiler
gcc-4.2-base	4.2.4-6	The GNU Compiler Collection (base package)
gcc-4.3	4.3.2-1.1	The GNU C compiler
gcc-4.3-base	4.3.2-1.1	The GNU Compiler Collection (base package)
gconf2	2.22.0-1	GNOME configuration database system (support tools)
gconf2-common	2.22.0-1	GNOME configuration database system (common files)
gdb	6.8-3	The GNU Debugger
gedit	2.22.3-1+lenny1	official text editor of the GNOME desktop environment
gedit-common	2.22.3-1+lenny1	official text editor of the GNOME desktop environment
genisoimage	9:1.1.9-1	Creates ISO-9660 CD-ROM filesystem images
gettext-base	0.17-4	GNU Internationalization utilities for the base system
gksu	2.0.0-8	graphical frontend to su
gnome-about	2.22.3-2	The GNOME about box
gnome-applets	2.22.3-3	Various applets for GNOME 2 panel - binary files
gnome-applets-data	2.22.3-3	Various applets for GNOME 2 panel - data files
gnome-control-center	1:2.22.2.1-2	utilities to configure the GNOME desktop
gnome-desktop-data	2.22.3-2	Common files for GNOME 2 desktop apps
gnome-doc-utils	0.12.2-1	a collection of documentation utilities for the Gnome project
gnome-icon-theme	2.22.0-1	GNOME Desktop icon theme
gnome-keyring	2.22.3-2	GNOME keyring services (daemon and tools)
gnome-media	2.22.0-3	GNOME media utilities
gnome-media-common	2.22.0-3	GNOME media utilities - common files
gnome-menus	2.22.2-4	an implementation of the freedesktop menu specification for GN
gnome-mime-data	2.18.0-1	base MIME and Application database for GNOME.
gnome-mount	0.7-2	wrapper for (un)mounting and ejecting storage devices
gnome-netstatus-applet	2.12.1-2	Network status applet for GNOME 2
gnome-panel	2.20.3-5	launcher and docking facility for GNOME
gnome-panel-data	2.20.3-5	common files for the GNOME Panel
gnome-session	2.22.3-2	The GNOME 2 Session Manager
gnome-settings-daemon	2.22.2.1-2	GNOME settings daemon
gnome-system-monitor	2.22.3-1	Process viewer and system resource monitor for GNOME 2
gnome-terminal	2.22.3-3	The GNOME 2 terminal emulator application
gnome-terminal-data	2.22.3-3	Data files for the GNOME terminal emulator
gnome-user-guide	2.22.1-1	GNOME user's guide
gnome-utils	2.20.0.1-3	GNOME desktop utilities
gnupg	1.4.9-3+lenny1	GNU privacy guard - a free PGP replacement
gpgv	1.4.9-3+lenny1	GNU privacy guard - signature verification tool
gpicview	0.1.9-4+lenny1	lightweight image viewer
grep	2.5.3~dfsg-6	GNU grep, egrep and fgrep
groff-base	1.18.1.1-21	GNU troff text-formatting system (base system)

		components)
grub	0.97-47lenny2	GRand Unified Bootloader (Legacy version)
grub-common	1.96+20080724-16	GRand Unified Bootloader, version 2 (common files)
gststreamer0.10-alsa	0.10.19-2	GStreamer plugin for ALSA
gststreamer0.10-plugins-base	0.10.19-2	GStreamer plugins from the "base" set
gststreamer0.10-plugins-good	0.10.8-4.1~lenny2	GStreamer plugins from the "good" set
gststreamer0.10-x	0.10.19-2	GStreamer plugins for X11 and Pango
gtk2-engines	1:2.14.3-2	theme engines for GTK+ 2.x
gzip	1.3.12-6	The GNU compression utility
hal	0.5.11-8	Hardware Abstraction Layer
hal-info	20080508+git20080601-1	Hardware Abstraction Layer - fdi files
hicolor-icon-theme	0.10-1	default fallback theme for FreeDesktop.org icon themes
hostname	2.95	utility to set/show the host name or domain name
iceweasel	3.0.6-3	lightweight web browser based on Mozilla
iegd-debian502	10.2.0	Moxa V2100 IEGD package
ifenslave	2	Attach and detach slave interfaces to a bonding device
ifenslave-2.6	1.1.0-10	Attach and detach slave interfaces to a bonding device
ifupdown	0.6.8+nmu1	high level tools to configure network interfaces
info	4.11.dfsg.1-4	Standalone GNU Info documentation browser
initramfs-tools	0.92o	tools for generating an initramfs
initscripts	2.86.ds1-61	Scripts for initializing and shutting down the system
installation-report	2.38	system installation report
iproute	20080725-2	networking and traffic control tools
iptables	1.4.2-6	administration tools for packet filtering and NAT
iputils-ping	3:20071127-1	Tools to test the reachability of network hosts
klibc-utils	1.5.12-2	small utilities built with klibc for early boot
laptop-detect	0.13.6	attempt to detect a laptop
leafpad	0.8.13-1	GTK+ based simple text editor
libaa1	1.4p5-37+b1	ascii art library
libacl1	2.2.47-2	Access control list shared library
libao2	0.8.8-4	Cross Platform Audio Output Library
libapache2-mod-php5	5.2.6.dfsg.1-1+lenny4	server-side, HTML-embedded scripting language (Apache 2 module)
libapm1	3.2.2-12	Library for interacting with APM driver in kernel
libapr1	1.2.12-5+lenny1	The Apache Portable Runtime Library
libaprutil1	1.2.12+dfsg-8+lenny4	The Apache Portable Runtime Utility Library
libapt-pkg-perl	0.1.22+b1	Perl interface to libapt-pkg
libart-2.0-2	2.3.20-2	Library of functions for 2D graphics - runtime files
libasound2	1.0.16-2	ALSA library
libaspell15	0.60.6-1	GNU Aspell spell-checker runtime library
libatk1.0-0	1.22.0-1	The ATK accessibility toolkit
libatk1.0-data	1.22.0-1	Common files for the ATK accessibility toolkit
libattr1	1:2.4.43-2	Extended attribute shared library
libaudiofile0	0.2.6-7	Open-source version of SGI's audiofile library
libavahi-client3	0.6.23-3lenny1	Avahi client library
libavahi-common-data	0.6.23-3lenny1	Avahi common data files

libavahi-common3	0.6.23-3lenny1	Avahi common library
libavahi-glib1	0.6.23-3lenny1	Avahi glib integration library
libavc1394-0	0.5.3-1+b1	control IEEE 1394 audio/video devices
libbeagle1	0.3.5-1+b1	library for accessing beagle using C
libbeecrypt6	4.1.2-7	open source C library of cryptographic algorithms
libblkid1	1.41.3-1	block device id library
libbonobo2-0	2.22.0-1	Bonobo CORBA interfaces library
libbonobo2-common	2.22.0-1	Bonobo CORBA interfaces library -- support files
libbonoboui2-0	2.22.0-1	The Bonobo UI library
libbonoboui2-common	2.22.0-1	The Bonobo UI library -- common files
libbz2-1.0	1.0.5-1	high-quality block-sorting file compressor library - runtime
libc6	2.7-18	GNU C Library: Shared libraries
libc6-dev	2.7-18	GNU C Library: Development Libraries and Header Files
libc6-i686	2.7-18	GNU C Library: Shared libraries [i686 optimized]
libcaca0	0.99.beta14-1	colour ASCII art library
libcairo-perl	1.060-1	Perl interface to the Cairo graphics library
libcairo2	1.6.4-7	The Cairo 2D vector graphics library
libcaiomm-1.0-1	1.6.0-1	C++ wrappers for Cairo (shared libraries)
libcamel1.2-11	2.22.3-1.1+lenny2	The Evolution MIME message handling library
libcap1	1:1.10-14	support for getting/setting POSIX.1e capabilities
libcap2	2.11-2	support for getting/setting POSIX.1e capabilities
libcdio7	0.78.2+dfsg1-3	library to read and control CD-ROM
libcdparanoia0	3.10.2+debian-5	audio extraction tool for sampling CDs (library)
libcomerr2	1.41.3-1	common error description library
libcompress-raw-zlib-perl	2.012-1lenny1	low-level interface to zlib compression library
libcompress-zlib-perl	2.012-1	Perl module for creation and manipulation of gzip files
libconfig-file-perl	1.42-1	Parses simple configuration files
libconsole	1:0.2.3dbs-65.1	Shared libraries for Linux console and font manipulation
libcpufreq0	004-2	shared library to deal with the cpufreq Linux kernel feature
libcroco3	0.6.1-2	a generic Cascading Style Sheet (CSS) parsing and manipulation
libcucul0	0.99.beta14-1	low-level Unicode character drawing library
libcups2	1.3.8-1+lenny6	Common UNIX Printing System(tm) - libs
libcwidget3	0.5.12-4	high-level terminal interface library for C++ (runtime files)
libdatrie0	0.1.3-2	Double-array trie library
libdb4.5	4.5.20-13	Berkeley v4.5 Database Libraries [runtime]
libdb4.6	4.6.21-11	Berkeley v4.6 Database Libraries [runtime]
libdbus-1-3	1.2.1-5+lenny1	simple interprocess messaging system
libdbus-glib-1-2	0.76-1 librar	simple interprocess messaging system (GLib-based shared)
libdevmapper1.02.1	2:1.02.27-4	The Linux Kernel Device Mapper userspace library
libdirectfb-1.0-0	1.0.1-11	direct frame buffer graphics - shared libraries
libdirectfb-extra	1.0.1-11	direct frame buffer graphics - extra providers
libdmx1	1:1.0.2-3	X11 Distributed Multihead extension library
libdrm2	2.3.1-2	Userspace interface to kernel DRM services -- runtime
libdv4	1.0.0-1+b1	software library for DV format digital video

		(runtime lib)
libebook1.2-9	2.22.3-1.1+lenny2	Client library for evolution address books
libecal1.2-7	2.22.3-1.1+lenny2	Client library for evolution calendars
libedata-book1.2-2	2.22.3-1.1+lenny2	Backend library for evolution address books
libedata-cal1.2-6	2.22.3-1.1+lenny2	Backend library for evolution calendars
libedataserver1.2-9	2.22.3-1.1+lenny2	Utility library for evolution data servers
libedataserverui1.2-8	2.22.3-1.1+lenny2	GUI utility library for evolution data servers
libedit2	2.11~20080614-1	BSD editline and history libraries
libeel2-2.20	2.20.0-7	Eazel Extensions Library (for GNOME2)
libeel2-data	2.20.0-7	Eazel Extensions Library - data files (for GNOME2)
libegroupwise1.2-13	2.22.3-1.1+lenny2	Client library for accessing groupwise POA through SOAP, interf
libenchant1c2a	1.4.2-3.3	a wrapper library for various spell checker engines
libept0	0.5.22	High-level library for managing Debian package information
libesd0	0.2.36-3	Enlightened Sound Daemon - Shared libraries
libevent1	1.3e-3	An asynchronous event notification library
libexempi3	2.0.1-1	library to parse XMP metadata (Library)
libexif12	0.6.16-2.1	library to parse EXIF files
libexpat1	2.0.1-4	XML parsing C library - runtime library
libfam0	2.7.0-13.3	Client library to control the FAM daemon
libffi5	3.0.7-1	Foreign Function Interface library runtime
libflac++6	1.2.1-1.2	Free Lossless Audio Codec - C++ runtime library
libflac8	1.2.1-1.2	Free Lossless Audio Codec - runtime C library
libfont-afm-perl	1.20-1	Font::AFM - Interface to Adobe Font Metrics files
libfontconfig1	2.6.0-3	generic font configuration library - runtime
libfontenc1	1:1.0.4-3	X11 font encoding library
libfreebob0	1.0.7-1	FreeBoB API
libfreetype6	2.3.7-2+lenny1	FreeType 2 font engine, shared library files
libfreezethaw-perl	0.43-4	converting Perl structures to strings and back
libfs6	2:1.0.1-1	X11 Font Services library
libgail-common modules	1.22.3-1	GNOME Accessibility Implementation Library -- common
libgail18	1.22.3-1	GNOME Accessibility Implementation Library -- shared library
libgamin0	0.1.9-2	Client library for the gamin file and directory monitoring sys
libgcc1	1:4.3.2-1.1	GCC support library
libgconf2-4	2.22.0-1	GNOME configuration database system (shared libraries)
libgcrypt11	1.4.1-1	LGPL Crypto library - runtime library
libgdata-google1.2-1	2.22.3-1.1+lenny2	Client library for accessing Google POA through SOAP interface
libgdata1.2-1	2.22.3-1.1+lenny2	Client library for accessing Google POA through SOAP interface
libgdbm3	1.8.3-3	GNU dbm database routines (runtime version)
libgksu2-0	2.0.7-1	library providing su and sudo functionality
libgl1-mesa-dri	7.0.3-7	A free implementation of the OpenGL API -- DRI modules
libgl1-mesa-glx	7.0.3-7	A free implementation of the OpenGL API -- GLX runtime
libglade2-0	1:2.6.2-1	library to load .glade files at runtime
libglib-perl	1:1.190-2	Perl interface to the GLib and GObject libraries

libglib2.0-0	2.16.6-2	The GLib library of C routines
libglib2.0-data	2.16.6-2	Common files for GLib library
libglibmm-2.4-1c2a	2.16.4-1	C++ wrapper for the GLib toolkit (shared libraries)
libglu1-mesa	7.0.3-7	The OpenGL utility library (GLU)
libgmp3c2	2:4.2.2+dfsg-3	Multiprecision arithmetic library
libgnome-desktop-2	2.22.3-2	Utility library for loading .desktop files - runtime files
libgnome-keyring0	2.22.3-2	GNOME keyring services library
libgnome-media0	2.22.0-3	runtime libraries for the GNOME media utilities
libgnome-menu2	2.22.2-4	an implementation of the freedesktop menu specification for GN
libgnome-window-settings1	1:2.22.2.1-2	Utility library for getting window manager settings
libgnome2-0	2.20.1.1-1	The GNOME 2 library - runtime files
libgnome2-canvas-perl	1.002-1+b2	Perl interface to the GNOME canvas library
libgnome2-common	2.20.1.1-1	The GNOME 2 library - common files
libgnome2-perl	1.042-1+b1	Perl interface to the GNOME libraries
libgnome2-vfs-perl	1.080-1+b1	Perl interface to the 2.x series of the GNOME VFS library
libgnomecanvas2-0	2.20.1.1-1	A powerful object-oriented display - runtime files
libgnomecanvas2-common	2.20.1.1-1	A powerful object-oriented display - common files
libgnomecups1.0-1	0.2.3-3	GNOME library for CUPS interaction
libgnomekbd-common files	2.22.0-1	GNOME library to manage keyboard configuration - common
libgnomekbd2 librar	2.22.0-1	GNOME library to manage keyboard configuration - shared
libgnomekbdui2	2.22.0-1	User interface library for libgnomekbd - shared library
libgnomeprint2.2-0	2.18.5-1	The GNOME 2.2 print architecture - runtime files
libgnomeprint2.2-data	2.18.5-1	The GNOME 2.2 print architecture - data files
libgnomeprintui2.2-0	2.18.3-1	GNOME 2.2 print architecture User Interface - runtime files
libgnomeprintui2.2-common	2.18.3-1	GNOME 2.2 print architecture User Interface - common files
libgnomeui-0	2.20.1.1-2	The GNOME 2 libraries (User Interface) - runtime files
libgnomeui-common	2.20.1.1-2	The GNOME 2 libraries (User Interface) - common files
libgnomevfs2-0	1:2.22.0-5	GNOME Virtual File System (runtime libraries)
libgnomevfs2-common	1:2.22.0-5	GNOME Virtual File System (common files)
libgnomevfs2-extra	1:2.22.0-5	GNOME Virtual File System (extra modules)
libgnutls26	2.4.2-6+lenny1	the GNU TLS library - runtime library
libgomp1	4.3.2-1.1	GCC OpenMP (GOMP) support library
libgpg-error0	1.4-2	library for common error values and messages in GnuPG component
libgpm2	1.20.4-3.1	General Purpose Mouse - shared library
libgsf-1-114	1.14.8-1lenny2	Structured File Library - runtime version
libgsf-1-common	1.14.8-1lenny2	Structured File Library - common files
libgssglue1	0.1-2	mechanism-switch gssapi library
libgstreamer-plugins-base0.10-0	0.10.19-2	GStreamer libraries from the "base" set
libgstreamer0.10-0	0.10.19-3	Core GStreamer libraries and elements
libgtk2-perl	1:1.190-1	Perl interface to the 2.x series of the Gimp Toolkit library
libgtk2.0-0	2.12.12-1~lenny1	The GTK+ graphical user interface library

libgtk2.0-bin	2.12.12-1~lenny1	The programs for the GTK+ graphical user interface library
libgtk2.0-common	2.12.12-1~lenny1	Common files for the GTK+ graphical user interface library
libgtkmm-2.4-1c2a	1:2.12.7-1	C++ wrappers for GTK+ 2.4 (shared libraries)
libgtksourceview-common	1.8.5-1	common files for the GTK+ syntax highlighting widget
libgtksourceview1.0-0	1.8.5-1	shared libraries for the GTK+ syntax highlighting widget
libgtksourceview2.0-0	2.2.2-1	shared libraries for the GTK+ syntax highlighting widget
libgtop2-7	2.22.3-1	gtop system monitoring library
libgtop2-common	2.22.3-1	common files for the gtop system monitoring library
libgucharmap6	1:2.22.3-2	Unicode browser widget library (shared library)
libgweather-common	2.22.3-1	GWeather common files
libgweather1	2.22.3-1	GWeather shared library
libhal-storage1	0.5.11-8	Hardware Abstraction Layer - shared library for storage device
libhal1	0.5.11-8	Hardware Abstraction Layer - shared library
libhtml-format-perl	2.04-2	format HTML syntax trees into text, PostScript or RTF
libhtml-parser-perl	3.56-1+b1	A collection of modules that parse HTML text documents
libhtml-tagset-perl	3.20-2	Data tables pertaining to HTML
libhtml-tree-perl	3.23-1	represent and create HTML syntax trees
libhunspell-1.2-0	1.2.6-1	spell checker and morphological analyzer (shared library)
libice6	2:1.0.4-1	X11 Inter-Client Exchange library
libid3tag0	0.15.1b-10	ID3 tag reading library from the MAD project
libidl0	0.8.10-0.1	library for parsing CORBA IDL files
libiec61883-0	1.1.0-2	an partial implementation of IEC 61883
libio-compress-base-perl	2.012-1	Base Class for IO::Compress modules
libio-compress-zlib-perl	2.012-1	Perl interface to zlib
libjack0	0.109.2-5	JACK Audio Connection Kit (libraries)
libjpeg-progs	6b-14	Programs for manipulating JPEG files
libjpeg62	6b-14	The Independent JPEG Group's JPEG runtime library
libkeyutils1	1.2-9	Linux Key Management Utilities (library)
libklibc	1.5.12-2	minimal libc subset for use with initramfs
libkrb53	1.6.dfsg.4~beta-5	lenny1 MIT Kerberos runtime libraries
liblcms1	1.17.dfsg-1+lenny 2	Color management library
libldap-2.4-2	2.4.11-1	OpenLDAP libraries
liblist-moreutils-perl	0.22-1+b1	Addition list functions not found in List::Util
liblocale-gettext-perl	1.05-4	Using libc functions for internationalization in Perl
liblockfile1	1.08-3	NFS-safe locking library, includes dotlockfile program
liblzo2-2	2.03-1	data compression library
libmad0	0.15.1b-4	MPEG audio decoder library
libmagic1	4.26-1	File type determination library using "magic" numbers
libmailtools-perl	2.03-1	Manipulate email in perl programs

libmalaga7	7.12-1	An automatic language analysis library
libmetacity0	1:2.22.0-2	library of lightweight GTK2 based Window Manager
libmldb-perl	2.01-2	Store multidimensional hash structures in perl tied hashes
libmozjs1d	1.9.0.14-0lenny1	The Mozilla SpiderMonkey JavaScript library
libmpfr1ldbl	2.3.1.dfsg.1-2	multiple precision floating-point computation
libmysqlclient15off	5.0.51a-24+lenny2	MySQL database client library
libnautilus-burn4	2.20.0-1	Nautilus Burn Library - runtime version
libnautilus-extension1	2.20.0-7	libraries for nautilus components - runtime version
libncurses5	5.7+20081213-1	shared libraries for terminal handling
libncursesw5	5.7+20081213-1	shared libraries for terminal handling (wide character support)
libncursesw5-dev	5.7+20081213-1	developer's libraries for ncursesw
libneon27	0.28.2-6.1	An HTTP and WebDAV client library
libnet-dbus-perl	0.33.6-1+b1	Extension for the Dbus bindings
libnet-lite-ftp-perl	0.54-2	Perl FTP client with support for TLS
libnet-ssleay-perl	1.35-1	Perl module for Secure Sockets Layer (SSL)
libnet-telnet-perl	3.03-3	Script telnetable connections
libnewt0.52	0.52.2-11.3	Not Erik's Windowing Toolkit - text mode windowing with slang
libnfsidmap2	0.20-1	An nfs idmapping library
libnotify1	0.4.4-3	sends desktop notifications to a notification daemon
libnspr4-0d	4.7.1-4	NetScape Portable Runtime Library
libnss3-1d	3.12.3.1-0lenny1	Network Security Service libraries
libobparser21	3.4.7.2-3	parsing library for openbox
libobrender21	3.4.7.2-3	rendering library for openbox themes
libogg0	1.1.3-4	Ogg Bitstream Library
liboil0.3	0.3.15-1	Library of Optimized Inner Loops
liboobs-1-4	2.22.0-2	GObject based interface to system-tools-backends - shared libr
liborbit2	1:2.14.13-0.1	libraries for ORBit2 - a CORBA ORB
libpam-gnome-keyring	2.22.3-2	PAM module to unlock the GNOME keyring upon login
libpam-modules	1.0.1-5+lenny1	Pluggable Authentication Modules for PAM
libpam-runtime	1.0.1-5+lenny1	Runtime support for the PAM library
libpam0g	1.0.1-5+lenny1	Pluggable Authentication Modules library
libpanel-applet2-0	2.20.3-5	library for GNOME Panel applets
libpango1.0-0	1.20.5-5	Layout and rendering of internationalized text
libpango1.0-common	1.20.5-5	Modules and configuration files for the Pango
libpcap0.8	0.9.8-5	system interface for user-level packet capture
libpci3	1:3.0.0-6	Linux PCI Utilities (shared library)
libpcre3	7.6-2.1	Perl 5 Compatible Regular Expression Library - runtime files
libperl5.10	5.10.0-19lenny2	Shared Perl library
libpixman-1-0	0.10.0-2	pixel-manipulation library for X and cairo
libpkcs11-helper1	1.05-1	library that simplifies the interaction with PKCS#11
libpng12-0	1.2.27-2+lenny2	PNG library - runtime
libpopt0	1.14-4	lib for parsing cmdline parameters
libpq5	8.3.7-0lenny1	PostgreSQL C client library
librarian0	0.8.1-1	Rarian is a documentation meta-data library (library package)

libraw1394-8	1.3.0-4	library for direct access to IEEE 1394 bus (aka FireWire)
libreadline5	5.2-3.1	GNU readline and history libraries, run-time libraries
librpcsecgss3	0.18-1	allows secure rpc communication using the rpcsec_gss protocol
librpm4.4	4.4.2.3-1	RPM shared library
librsvg2-2	2.22.2-2lenny1	SAX-based renderer library for SVG files (runtime)
librsvg2-common	2.22.2-2lenny1	SAX-based renderer library for SVG files (extra runtime)
libsasl2-2	2.1.22.dfsg1-23+lenny1	Cyrus SASL - authentication abstraction library
libscrollkeeper0	0.3.14-16	Library to load .omf files (runtime files)
libselenium1	2.0.65-5	SELinux shared libraries
libsensors3	1:2.10.7-1	library to read temperature/voltage/fan sensors
libsepol1	2.0.30-2	Security Enhanced Linux policy library for changing policy bin
libsexy2	0.1.11-2+b1	collection of additional GTK+ widgets - library
libshout3	2.2.2-5	MP3/Ogg Vorbis broadcast streaming library
libsigc++-2.0-0c2a	2.0.18-2	type-safe Signal Framework for C++ - runtime
libslab0	0.9.8.svn.20070430-1.1	beautification app libraryfile
libslang2	2.1.3-3	The S-Lang programming library - runtime version
libsm6	2:1.0.3-2	X11 Session Management library
libsmbclient	2:3.2.5-4lenny6	shared library that allows applications to talk to SMB/CIFS se
libsmbios-bin	2.0.3.dfsg-1	Provide access to (SM)BIOS information -- utility binaries
libsmbios2	2.0.3.dfsg-1	Provide access to (SM)BIOS information -- dynamic library
libsndfile1	1.0.17-4+lenny2	Library for reading/writing audio files
libsnmp-base	5.4.1~dfsg-12	SNMP (Simple Network Management Protocol) MIBs and documentation
libsnmp15	.4.1~dfsg-12	SNMP (Simple Network Management Protocol) library
libsoundtouch1c2	.3.1-2	sound stretching library
libsoup2.4-1	2.4.1-2	an HTTP library implementation in C -- Shared library
libspeex1	1.2~rc1-1	The Speex codec runtime library
libsplashy1	0.3.13-3	Library to draw splash screen on boot, shutdown, resume or sus
libsqlite3-0	3.5.9-6	SQLite 3 shared library
libss2	1.41.3-1	command-line interface parsing library
libssl0.9.8	0.9.8g-15+lenny5	SSL shared libraries
libstartup-notification0	0.9-1	library for program launch feedback (shared library)
libstdc++6	4.3.2-1.1	The GNU Standard C++ Library v3
libstdc++6-4.3-dev	4.3.2-1.1	The GNU Standard C++ Library v3 (development files)
libsvga1	1:1.4.3-27	console SVGA display libraries
libsysfs2	2.1.0-5	interface library to sysfs
libtag1c2a	1.5-3	TagLib Audio Meta-Data Library
libtalloc1	1.2.0~git2008061	hierarchical pool based memory allocator



	6-1	
libtasn1-3	1.4-1	Manage ASN.1 structures (runtime)
libtext-charwidth-perl	0.04-5+b1	get display widths of characters on the terminal
libtext-iconv-perl	1.7-1+b1	converts between character sets in Perl
libtext-wrapi18n-perl	0.06-6	internationalized substitute of Text::Wrap
libthai-data	0.1.9-4	Data files for Thai language support library
libthai0	0.1.9-4	Thai language support library
libtheora0	1.0~beta3-1	The Theora Video Compression Codec
libtie-ixhash-perl	1.21-2	ordered associative arrays for Perl
libtiff4	3.8.2-11.2	Tag Image File Format (TIFF) library
libtimedate-perl	1.1600-9	Time and date functions for Perl
libtotem-plparser10	2.22.3-1	Totem Playlist Parser library - runtime version
libtrackerclient0	0.6.6-2	metadata database, indexer and search tool - library
libts-0.0-0	1.0-4	touch screen library
liburi-perl	1.35.dfsg.1-1	Manipulates and accesses URI strings
libusb-0.1-4	2:0.1.12-13	userspace USB programming library
libuuid-perl	0.02-3+b1	Perl extension for using UUID interfaces as defined in e2fspro
libuuid1	1.41.3-1	universally unique id library
libvisual-0.4-0	0.4.0-2.1	Audio visualization framework
libvisual-0.4-plugins	0.4.0.dfsg.1-2	Audio visualization framework plugins
libvoikko1	1.7-2	Finnish spell-checker and hyphenator library
libvolume-id0	0.125-7+lenny3	libvolume_id shared library
libvorbis0a	1.2.0.dfsg-3.1	The Vorbis General Audio Compression Codec
libvorbisenc2	1.2.0.dfsg-3.1	The Vorbis General Audio Compression Codec
libvorbisfile3	1.2.0.dfsg-3.1	The Vorbis General Audio Compression Codec
libvte-common	1:0.16.14-4	Terminal emulator widget for GTK+ 2.0 - common files
libvte9	1:0.16.14-4	Terminal emulator widget for GTK+ 2.0 - runtime files
libwavpack1	4.50.1-1	an audio codec (lossy and lossless) - library
libwbclient0	2:3.2.5-4+lenny6	client library for interfacing with winbind service
libwnck-common	2.22.3-1	Window Navigator Construction Kit - common files
libwnck22	2.22.3-1	Window Navigator Construction Kit - runtime files
libwrap0	7.6.q-16	Wietse Venema's TCP wrappers library
libwww-perl	5.813-1	WWW client/server library for Perl (aka LWP)
libwxbase2.6-0	2.6.3.2.2-3+lenny1	wxBase library (runtime) - non-GUI support classes of wxWidget
libwxgtk2.6-0	2.6.3.2.2-3+lenny1	wxWidgets Cross-platform C++ GUI toolkit (GTK+ runtime)
libx11-6	2:1.1.5-2	X11 client-side library
libx11-data	2:1.1.5-2	X11 client-side library
libx86-1	1.1+ds1-2	x86 real-mode library
libxapian15	1.0.7-4	Search engine library
libxau6	1:1.0.3-3	X11 authorisation library
libxaw7	2:1.0.4-2	X11 Athena Widget library
libxcb-render-util0	0.2.1+git1-1	utility libraries for X C Binding -- render-util
libxcb-render0	1.1-1.2	X C Binding, render extension
libxcb-xlib0	1.1-1.2	X C Binding, Xlib/XCB interface library
libxcb1	1.1-1.2	X C Binding
libxcomposite1	1:0.4.0-3	X11 Composite extension library
libxcursor1	1:1.1.9-1	X cursor management library

libxdamage1	1:1.1.1-4	X11 damaged region extension library
libxdmcp6	1:1.0.2-3	X11 Display Manager Control Protocol library
libxext6	2:1.0.4-1	X11 miscellaneous extension library
libxfixed3	1:4.0.3-2	X11 miscellaneous 'fixes' extension library
libxfont1	1:1.3.3-1	X11 font rasterisation library
libxft2	2.1.12-3	FreeType-based font drawing library for X
libxi6	2:1.1.4-1	X11 Input extension library
libxinerama1	2:1.0.3-2	X11 Xinerama extension library
libxkbfile1	1:1.0.5-1	X11 keyboard file manipulation library
libxklavier12	3.5-2	X Keyboard Extension high-level API
libxml-parser-perl	2.36-1.1+b1	Perl module for parsing XML files
libxml-twig-perl	1:3.32-1	Perl module for processing huge XML documents in tree mode
libxml-xpath-perl	1.13-6	Perl module for processing XPath
libxml2	2.6.32.dfsg-5+len ny1	GNOME XML library
libxml2-utils	2.6.32.dfsg-5+len ny1	XML utilities
libxmu6	2:1.0.4-1	X11 miscellaneous utility library
libxmuu1	2:1.0.4-1	X11 miscellaneous micro-utility library
libxpm4	1:3.5.7-1	X11 pixmap library
libxrandr2	2:1.2.3-1	X11 RandR extension library
libxrender1	1:0.9.4-2	X Rendering Extension client library
libxres1	2:1.0.3-1	X11 Resource extension library
libxslt1.1	1.1.24-2	XSLT processing library - runtime library
libxss1	1:1.1.3-1	X11 Screen Saver extension library
libxt6	1:1.0.5-3	X11 toolkit intrinsics library
libxtrap6	2:1.0.0-5	X11 event trapping extension library
libxtst6	2:1.0.3-1	X11 Testing -- Resource extension library
libxv1	2:1.0.4-1	X11 Video extension library
libxxf86dga1	2:1.0.2-1	X11 Direct Graphics Access extension library
libxxf86misc1	1:1.0.1-3	X11 XFree86 miscellaneous extension library
libxxf86vm1	1:1.0.2-1	X11 XFree86 video mode extension library
linux-image-2.6.26	moxa.1.0	Linux kernel binary image for version 2.6.26
linux-libc-dev	2.6.26-19	Linux support headers for userspace development
linux-sound-base	1.0.17.dfsg-4	base package for ALSA and OSS sound systems
locales	2.7-18	GNU C Library: National Language (locale) data [support]
lockfile-progs	0.1.11-0.1	Programs for locking and unlocking files and mailboxes
login	1:4.1.1-6	system login tools
logrotate	3.7.1-5	Log rotation utility
lrzsz	0.12.21-4.1	Tools for zmodem/xmodem/ymodem file transfer
lsb-base	3.2-20	Linux Standard Base 3.2 init script functionality
lsdf	4.78.dfsg.1-4	List open files
lxappearance	0.2-1	a new feature-rich GTK+ theme switcher
lxde	0.3.2.1+svn20080 509-5	Meta-package for the Lightweight X11 Desktop Environment
lxde-common	0.3.2.1+svn20080 509-5	the Lightweight X11 Desktop Environment configuration data
lxde-core	0.3.2.1+svn20080 509-5	Meta-package for the Lightweight X11 Desktop Environment Core
lxde-settings-daemon	0.3.2.1+svn20080	LXDE settings daemon

	509-5	
lxpanel	0.3.8.1-2	a lightweight desktop panel for X
lxrandr	0.1+svn20080716-3	simple monitor config tool for LXDE
lxsession-lite	0.3.6-2a	lightweight X11 session manager (lite version)
lxterminal	0.1.3-2	desktop independent vte-based terminal emulator
lzma	4.43-14	Compression method of 7z format in 7-Zip program
m4	1.4.11-1	a macro processing language
make	3.81-5	The GNU version of the "make" utility.
mkdev	2.3.1-88	creates device files in /dev
man-db	2.5.2-4	on-line manual pager
manpages	3.05-1	Manual pages about using a GNU/Linux system
mawk	1.3.3-11.1	a pattern scanning and text processing language
menu	2.1.41	generates programs menu for all menu-aware applications
menu-xdg	0.3	freedesktop.org menu compliant window manager scripts
metacity	1:2.22.0-2	A lightweight GTK2 based Window Manager
metacity-common	1:2.22.0-2	Shared files of lightweight GTK2 based Window Manager
mime-support	3.44-1	MIME files 'mime.types' & 'mailcap', and support programs
minicom	2.3-1	friendly menu driven serial communication program
miscfiles	1.4.2.dfsg.1-9	Dictionaries and other interesting files
mktemp	1.5-9	tool for creating temporary files
modconf	0.3.9	Device Driver Configuration
module-init-tools	3.4-1	tools for managing Linux kernel modules
mount	2.13.1.1-1	Tools for mounting and manipulating filesystems
myspell-en-us	1:2.4.0-3	English_american dictionary for myspell
mysql-common	5.0.51a-24+lenny2	MySQL database common files
nano	2.0.7-4	free Pico clone with some new features
nautilus	2.20.0-7	file manager and graphical shell for GNOME
nautilus-cd-burner	2.20.0-1	CD Burning front-end for Nautilus
nautilus-data	2.20.0-7	data files for nautilus
ncurses-base	5.7+20081213-1	basic terminal type definitions
ncurses-bin	5.7+20081213-1	terminal-related programs and man pages
net-tools	1.60-22	The NET-3 networking toolkit
netbase	4.34	Basic TCP/IP networking system
netcat-traditional	1.10-38	TCP/IP swiss army knife
nfs-common	1:1.1.2-6lenny1	NFS support files common to client and server
notification-daemon	0.3.7-1+b1	a daemon that displays passive pop-up notifications
ntp	1:4.2.4p4+dfsg-8lenny2	Network Time Protocol daemon and utility programs
ntpdate	1:4.2.4p4+dfsg-8lenny2	client for setting system time from NTP servers
obconf	2.0.3-3	Preferences manager for Openbox window manager
openbox	3.4.7.2-3	standards compliant, fast, light-weight, extensible window man

openbox-themes	1.0.2	Themes for the Openbox window manager
openbsd-inetd	0.20080125-2	The OpenBSD Internet Superserver
openssh-blacklist	0.4.1	list of default blacklisted OpenSSH RSA and DSA keys
openssh-blacklist-extra	0.4.1	list of non-default blacklisted OpenSSH RSA and DSA keys
openssh-client	1:5.1p1-5	secure shell client, an rlogin/rsh/rcp replacement
openssh-server	1:5.1p1-5	secure shell server, an rshd replacement
openssl	0.9.8g-15+lenny5	Secure Socket Layer (SSL) binary and related cryptographic too
openssl-blacklist	0.4.2	list of blacklisted OpenSSL RSA keys
openvpn	2.1~rc11-1	virtual private network daemon
openvpn-blacklist	0.3	list of blacklisted OpenVPN RSA shared keys
oss-compat	0.0.4+nmu2	OSS compatibility package
p7zip-full	4.58~dfsg.1-1	7z and 7za file archivers with high compression ratio
passwd	1:4.1.1-6	change and administer password and group data
pciutils	1:3.0.0-6	Linux PCI Utilities
pcmanfm	0.5-3	an extremely fast and lightweight file manager for X
perl	5.10.0-19lenny2	Larry Wall's Practical Extraction and Report Language
perl-base	5.10.0-19lenny2	minimal Perl system
perl-modules	5.10.0-19lenny2	Core Perl modules
php5-common	5.2.6.dfsg.1-1+lenny4	Common files for packages built from the php5 source
pm-utils	1.1.2.4-1	utilities and scripts for power management
portmap	6.0-9	RPC port mapper
powermgmt-base	1.30+nmu1	Common utils and configs for power management
ppp	2.4.4rel-10.1	Point-to-Point Protocol (PPP) - daemon
pppconfig	2.3.18	A text menu based utility for configuring ppp
pppoe	3.8-3	PPP over Ethernet driver
pppoeconf	1.18	configures PPPoE/ADSL connections
procps	1:3.2.7-11	/proc file system utilities
proftpd	1.3.1-17lenny2	versatile, virtual-hosting FTP daemon
proftpd-basic	1.3.1-17lenny2	versatile, virtual-hosting FTP daemon - binaries
proftpd-mod-ldap	1.3.1-17lenny2	versatile, virtual-hosting FTP daemon - LDAP module
proftpd-mod-mysql	1.3.1-17lenny2	versatile, virtual-hosting FTP daemon - MySQL module
proftpd-mod-pgsql	1.3.1-17lenny2	versatile, virtual-hosting FTP daemon - PostgreSQL module
psmisc	22.6-1	Utilities that use the proc filesystem
python	2.5.2-3	An interactive high-level object-oriented language (default ve
python-beagle	0.3.5-1+b1	Python bindings for beagle
python-cairo	1.4.12-1.2	Python bindings for the Cairo vector graphics library
python-central	0.6.8	register and build utility for Python packages
python-dbus	0.82.4-2	simple interprocess messaging system (Python interface)
python-fpconst	0.7.2-4	Utilities for handling IEEE 754 floating point special values

python-glade2	2.12.1-6	GTK+ bindings: Glade support
python-gmenu	2.22.2-4	an implementation of the freedesktop menu specification for GN
python-gnome2	2.22.0-1	Python bindings for the GNOME desktop environment
python-gnome2-desktop	2.22.0-2	Python bindings for the GNOME desktop environment
python-gobject	2.14.2-2	Python bindings for the GObject library
python-gtk2	2.12.1-6	Python bindings for the GTK+ widget set
python-libxml2	2.6.32.dfsg-5+	lenny1 Python bindings for the GNOME XML library
python-minimal	2.5.2-3	A minimal subset of the Python language (default version)
python-numeric	24.2-9	Numerical (matrix-oriented) Mathematics for Python
python-pyorbit	2.14.3-2	A Python language binding for the ORBit2 CORBA implementation
python-soappy	0.12.0-4	SOAP Support for Python
python-support	0.8.4lenny1	automated rebuilding support for Python modules
python2.5	2.5.2-15	An interactive high-level object-oriented language (version 2.
python2.5-minimal	2.5.2-15	A minimal subset of the Python language (version 2.5)
radeontool	1.5-5	utility to control ATI Radeon backlight functions on laptops
readline-common	5.2-3.1	GNU readline and history libraries, common files
rpm	4.4.2.3-1	Red Hat package manager
rsyslog	3.18.6-4	enhanced multi-threaded syslogd
scrollkeeper	0.3.14-16	A free electronic cataloging system for documentation
sed	4.1.5-6	The GNU sed stream editor
sgml-base	1.26	SGML infrastructure and SGML catalog file support
sgml-data	2.0.3	common SGML and XML data
shared-mime-info	0.30-2	FreeDesktop.org shared MIME database and spec
snmp	5.4.1~dfsg-12	SNMP (Simple Network Management Protocol) applications
snmpd	5.4.1~dfsg-12	SNMP (Simple Network Management Protocol) agents
ssh	1:5.1p1-5	secure shell client and server (metapackage)
ssl-cert	1.0.23	simple debconf wrapper for OpenSSL
strace	4.5.17+cvs080723-2	A system call tracer
sudo	1.6.9p17-2	Provide limited super user privileges to specific users
synaptic	0.62.1+nmu1	Graphical package manager
system-tools-backends	2.6.0-2lenny3	System Tools to manage computer configuration -- scripts
sysv-	2.86.ds1-61	System-V-like runlevel change mechanism
sysvinit	2.86.ds1-61	System-V-like init utilities
sysvinit-utils	2.86.ds1-61	System-V-like utilities
tar	1.20-1	GNU version of the tar archiving utility
tasksel	2.78	Tool for selecting tasks for installation on Debian systems
tasksel-data	2.78	Official tasks used for installation of Debian

		systems
tcpd	7.6.q-16	Wietse Venema's TCP wrapper utilities
tcpdump	3.9.8-4	A powerful tool for network monitoring and data acquisition
telnet	0.17-36	The telnet client
telnetd	0.17-36	The telnet server
tftpd	0.17-16	Trivial file transfer protocol server
tightvncserver	1.3.9-4	virtual network computing server software
traceroute	2.0.11-2	Traces the route taken by packets over an IPv4/IPv6 network
ttf-dejavu	2.25-3	Metapackage to pull in ttf-dejavu-core and ttf-dejavu-extra
ttf-dejavu-core	2.25-3	Vera font family derivate with additional characters
ttf-dejavu-extra	2.25-3	Vera font family derivate with additional characters
tzdata	2009l-0lenny1	time zone and daylight-saving time data
ucf	3.0016	Update Configuration File: preserve user changes to config fil
udev	0.125-7+lenny3	/dev/ and hotplug management daemon
unzip	5.52-12	De-archiver for .zip files
update-inetd	4.31	inetd configuration file updater
usbmount	0.0.14.1	automatically mount and unmount USB mass storage devices
usbutils	0.73-10	Linux USB utilities
uswsusp	0.7-1.2	tools to use userspace software suspend provided by Linux
util-linux	2.13.1.1-1	Miscellaneous system utilities
vbetool	1.0-3	run real-mode video BIOS code to alter hardware state
vim	1:7.1.314-3+lenny2	Vi IMproved - enhanced vi editor
vim-common	1:7.1.314-3+lenny2	Vi IMproved - Common files
vim-runtime	1:7.1.314-3+lenny2	Vi IMproved - Runtime files
vim-tiny	1:7.1.314-3+lenny2	Vi IMproved - enhanced vi editor - compact version
vino	2.22.2-1	VNC server for GNOME
watchdog	5.4-10	A software watchdog
wget	1.11.4-2	retrieves files from the web
whiptail	0.52.2-11.3	Displays user-friendly dialog boxes from shell scripts
wodim	9:1.1.9-1	command line CD/DVD writing tool
x-ttcidfont-conf	31	TrueType and CID fonts configuration for X
x11-apps	7.3+4	X applications
x11-common	1:7.3+20	X Window System (X.Org) infrastructure
x11-session-utils	7.3+1	X session utilities
x11-utils	7.3+2+nmu1	X11 utilities
x11-xfutils	7.3+1	X font server utilities
x11-xkb-utils	7.4+1	X11 XKB utilities
x11-xserver-utils	7.3+5	X server utilities
xarchiver	0.4.6-8	GTK+ frontend for most used compression formats
xauth	1:1.0.3-2	X authentication utility
xbase-clients	1:7.3+20	miscellaneous X clients - metapackage

xbitmaps	1.0.1-2	Base X bitmaps
xdg-utils	1.0.2-6.1	desktop integration utilities from freedesktop.org
xfonts-100dpi	1:1.0.0-4	100 dpi fonts for X
xfonts-75dpi	1:1.0.0-4	75 dpi fonts for X
xfonts-base	1:1.0.0-5	standard fonts for X
xfonts-encodings	1:1.0.2-3	Encodings for X.Org fonts
xfonts-scalable	1:1.0.0-6	scalable fonts for X
xfonts-utils	1:7.4+1	X Window System font utility programs
xinit	1.0.9-2	X server initialisation tool
xkb-data	1.3-2	X Keyboard Extension (XKB) configuration data
xli	1.17.0+20061110-3	command line tool for viewing images in X11
xml-core	0.12	XML infrastructure and XML catalog file support
xorg	1:7.3+20	X.Org X Window System
xorg-docs	1:1.4-4	Miscellaneous documentation for the X.Org software suite
xscreensaver	5.05-3	Automatic screensaver for X
xscreensaver-data	5.05-3	data files to be shared among screensaver frontends
xserver-xorg	1:7.3+20	the X.Org X server
xserver-xorg-core	2:1.4.2-10.lenny2	Xorg X server - core server
xserver-xorg-input-all	1:7.3+20	the X.Org X server -- input driver metapackage
xserver-xorg-input-evdev	1:2.0.8-1	X.Org X server -- evdev input driver
xserver-xorg-input-kbd	1:1.3.1-1	X.Org X server -- keyboard input driver
xserver-xorg-input-mouse	1:1.3.0-1	X.Org X server -- mouse input driver
xserver-xorg-input-synaptics	0.14.7~git20070706-3	Synaptics TouchPad driver for X.Org/XFree86 server
xserver-xorg-input-wacom	0.7.9.3-2	X.Org X server -- Wacom input driver
xserver-xorg-video-all	1:7.3+20	the X.Org X server -- output driver metapackage
xserver-xorg-video-apm	1:1.2.0-1	X.Org X server -- APM display driver
xserver-xorg-video-ark	1:0.7.0-1	X.Org X server -- ark display driver
xserver-xorg-video-ati	1:6.9.0-1+lenny4	X.Org X server -- ATI display driver wrapper
xserver-xorg-video-chips	1:1.2.0-1	X.Org X server -- Chips display driver
xserver-xorg-video-cirrus	1:1.2.1-1.lenny1	X.Org X server -- Cirrus display driver
xserver-xorg-video-cyrix	1:1.1.0-8	X.Org X server -- Cyrix display driver
xserver-xorg-video-dummy	1:0.3.0-1	X.Org X server -- dummy display driver
xserver-xorg-video-fbdev	1:0.4.0-1	X.Org X server -- fbdev display driver
xserver-xorg-video-glint	1:1.2.1-1	X.Org X server -- Glint display driver
xserver-xorg-video-i128	1:1.3.0-1	X.Org X server -- i128 display driver
xserver-xorg-video-i740	1:1.2.0-1	X.Org X server -- i740 display driver
xserver-xorg-video-imstt	1:1.1.0-7	X.Org X server -- IMSTT display driver
xserver-xorg-video-intel	2:2.3.2-2+lenny6	X.Org X server -- Intel i8xx, i9xx display driver
xserver-xorg-video-mach64	6.8.0-1	X.Org X server -- ATI Mach64 display driver
xserver-xorg-video-mga	1:1.4.9.dfsg-1	X.Org X server -- MGA display driver
xserver-xorg-video-neomagic	1:1.2.1-1	X.Org X server -- Neomagic display driver
xserver-xorg-video-nsc	1:2.8.3-4	X.Org X server -- NSC Geode GX1 display driver
xserver-xorg-video-nv	1:2.1.10-1	X.Org X server -- NV display driver
xserver-xorg-video-openchrome	1:0.2.902+svn579-4	X.Org X server -- VIA display driver
xserver-xorg-video-r128	6.8.0-1	X.Org X server -- ATI r128 display driver
xserver-xorg-video-radeon	1:6.9.0-1+lenny4	X.Org X server -- ATI Radeon display driver
xserver-xorg-video-radeonhd	1.2.1-	X.Org X server -- AMD/ATI r5xx, r6xx display driver

xserver-xorg-video-rendition	1:4.20.dfsg.1-2	X.Org X server -- Rendition display driver
xserver-xorg-video-s3	1:0.6.0-1	X.Org X server -- legacy S3 display driver
xserver-xorg-video-s3virge	1:1.10.1-1	X.Org X server -- S3 ViRGE display driver
xserver-xorg-video-savage	1:2.2.1-2.lenny1	X.Org X server -- Savage display driver
xserver-xorg-video-siliconmotion	1:1.6.0-1	X.Org X server -- SiliconMotion display driver
xserver-xorg-video-sis	1:0.10.0-1	X.Org X server -- SiS display driver
xserver-xorg-video-sisusb	1:0.9.0-1	X.Org X server -- SiS USB display driver
xserver-xorg-video-tdfx	1:1.4.0-1	X.Org X server -- tdfx display driver
xserver-xorg-video-tga	1:1.1.0-9	X.Org X server -- TGA display driver
xserver-xorg-video-trident	1:1.3.0-1	X.Org X server -- Trident display driver
xserver-xorg-video-tseng	1:1.2.0-1	X.Org X server -- Tseng display driver
xserver-xorg-video-v4l	0.2.0-1	X.Org X server -- Video 4 Linux display driver
xserver-xorg-video-vesa	1:1.3.0-4	X.Org X server -- VESA display driver
xserver-xorg-video-vga	1:4.1.0-8	X.Org X server -- VGA display driver
xserver-xorg-video-vmware	1:10.16.2-1	X.Org X server -- VMware display driver
xserver-xorg-video-vooodoo	1:1.2.0-1	X.Org X server -- Voodoo display driver
xsltproc	1.1.24-2	XSLT command line processor
xterm	235-2	X terminal emulator
xulrunner-1.9	1.9.0.14-0lenny1	XUL + XPCOM application runner
yelp	2.22.1-8+b1	Help browser for GNOME 2
zenity	2.22.1-2	Display graphical dialog boxes from shell scripts
zip	2.32-1	Archiver for .zip files
zlib1g	1:1.2.3.3.dfsg-12	compression library - runtime



# Software Configuration

---

In this chapter, we explain how to operate a V2101-LX computer directly or your desktop. There are three ways to connect to the V2101-LX computer: through VGA monitor, by using Telnet over the network, or by using an SSH console from a Windows or Linux machine. This chapter describes basic Linux operating system configurations. The advanced network management and configuration will be described in the next chapter “Managing Communications.”

The following topics are covered in this chapter:

- ❑ **Starting from a VGA Console**
- ❑ **Connecting from a Telnet Console**
- ❑ **Connecting from an SSH Console**
  - Windows Users
  - Linux Users
- ❑ **Adjusting the System Time**
  - Setting the Time Manually
  - NTP Client
  - Updating the Time Automatically
- ❑ **Enabling and Disabling Daemons**
- ❑ **Setting the Run-Level**
- ❑ **Cron—Daemon for Executing Scheduled Commands**
- ❑ **Inserting a USB Storage Device into the Computer**
- ❑ **VGA & LVDS Configuration**
- ❑ **Checking the Linux Version**
- ❑ **APT—Installing and Removing Packages**

## Starting from a VGA Console

Connect the display monitor to the V2101-LX VGA connector, and then power it up by connecting it to the power adaptor. It takes about 30 to 60 seconds for the system to boot up. Once the system is ready, a login screen will appear on your monitor.

To log in, type the login name and password as requested. The default values are both **root**.

**Login: root**

**Password: root**

```
Moxa login: root
Password:
Last login: Mon Jan 22 19:02:16 2007 from 192.168.3.120

#####          #####          #####          #####          #####          ##
###            #####          ###          ###          #####          ###
###            ###          ###          ###          ###          ##          ###
###            #####          ##          ##          ###          #          #####
#####          #          ##          ###          ###          ##          #          ##
## ##          #          ##          ###          ##          #####          #          ##
## ##          ##          ##          ##          ##          #####          #          ###
##          #          ##          ##          ##          ##          #####          #####
##          #          ##          ##          ##          ##          #####          #          ##
##          ##          ##          ##          ##          ##          ##          #          ##
##          ##          ##          ##          ##          ##          ##          #          ##
##          ##          ##          ##          ##          #          ##          #          ##
#####          #          #####          #####          #####          #####          #####

For further information check:
http://www.moxa.com/
Mount user file system.

Moxa:~#
```

## Connecting from a Telnet Console

The V2101-LX computer comes with four basic Gigabit Ethernet ports named LAN1 and LAN2. The default IP addresses and netmasks of the network interfaces are as follows:

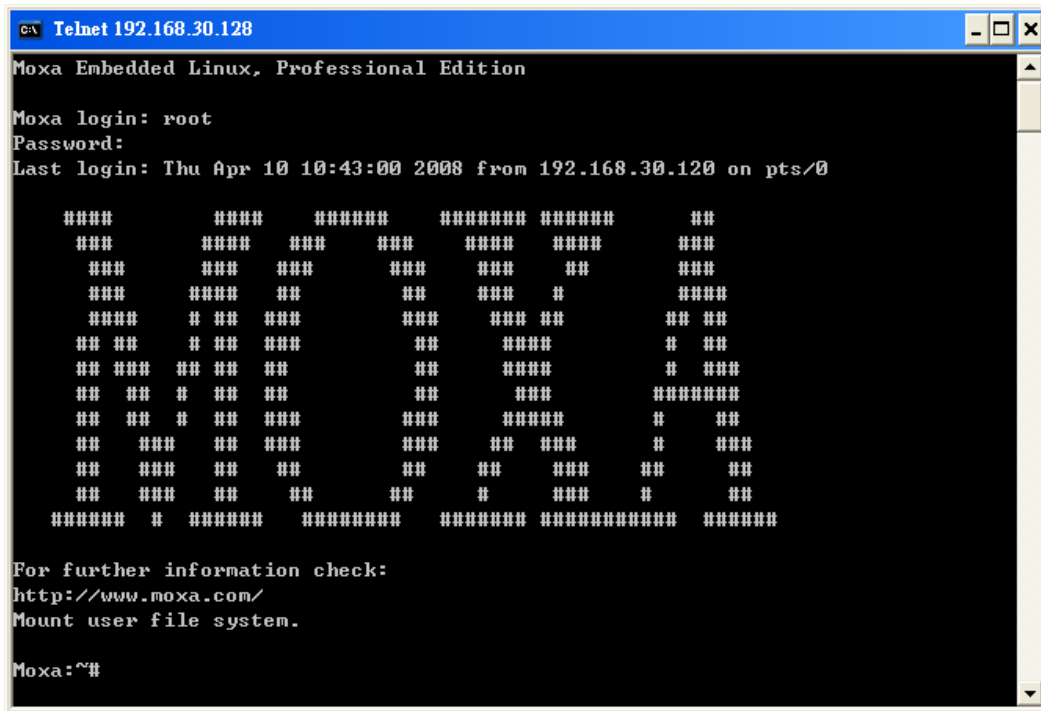
	Default IP Address	Netmask
LAN 1	192.168.3.127	255.255.255.0
LAN 2	192.168.4.127	255.255.255.0

Before using the Telnet client, you should change the IP address of your development workstation so that the network ports are on the same subnet as the IP address for the LAN port that you connect to. For example, if you connect to LAN 1, you could set your PC's IP address to 192.168.3.126, and the netmask to 255.255.255.0. If you connect to LAN 2, you can set your PC's IP address to 192.168.4.126, and the netmask to 255.255.255.0.

Use a cross-over Ethernet cable to connect your development workstation directly to the target computer, or use a straight-through Ethernet cable to connect the computer to a LAN hub or switch. Next, use a Telnet client on your development workstation to connect to the target computer. After a connection has been established, type the login name and password as requested to log on to the computer. The default values are both **root**.

**Login: root**

**Password: root**

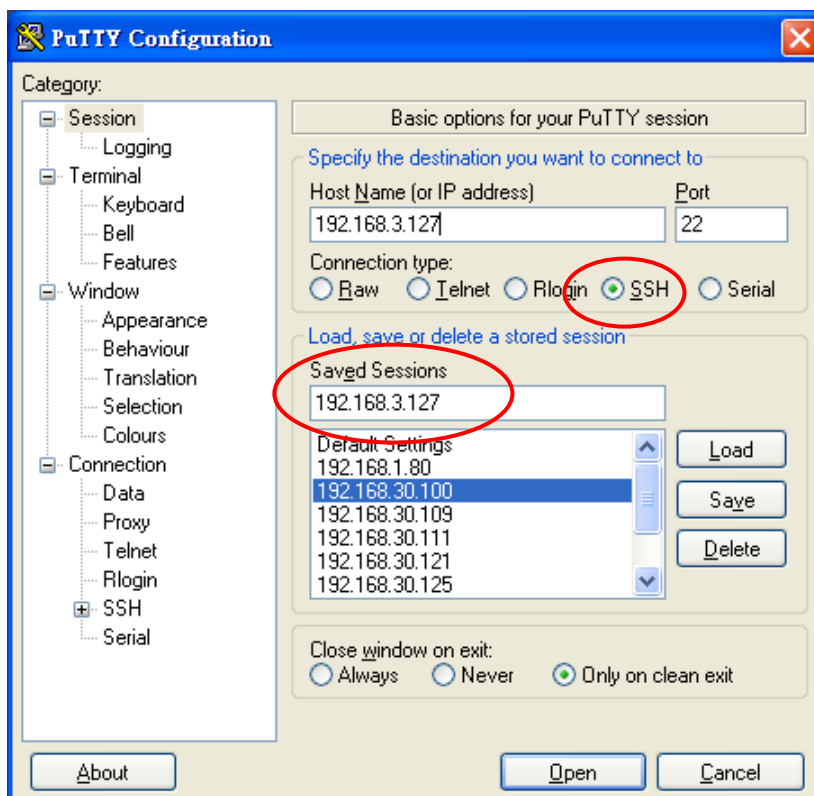


## Connecting from an SSH Console

The V2101-LX computer supports an SSH Console to offer users with better security over the network compared to Telnet.

## Windows Users

Click on the link <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> to download **PuTTY** (free software) to set up an SSH console for the V2101-LX in a Windows environment. The following screen shows an example of the configuration that is required.



## Linux Users

From a Linux machine, use the **ssh** command to access the V2101-LX's console utility via SSH.

```
#ssh 192.168.3.127
```

Select **yes** to open the connection.

```
[root@bee_notebook root]# ssh 192.168.3.127
The authenticity of host '192.168.3.127 (192.168.3.127)' can't be established.
RSA key fingerprint is 8b:ee:ff:84:41:25:fc:cd:2a:f2:92:8f:cb:1f:6b:2f.
Are you sure you want to continue connection (yes/no)? yes_
```

## Adjusting the System Time

The V2101-LX has two time settings. One is the system time, and the other is provided by an RTC (Real Time Clock) built into the V2101-LX's hardware.

### Setting the Time Manually

Use the **date** command to query the current system time or set a new system time. Use **hwclock** to query the current RTC time or set a new RTC time.

Use the following command to set the system time.

```
moxa@moxa:~# date MMDDhhmmYYYY
```

```
MM:      Month
DD:      Date
hhmm:    Hour and Minute
YYYY:    Year
```

Use the following command to write the current system time to the RTC.

```
moxa@moxa:~# hwclock -w
```

```
MOXA:~# date
Wed Dec 16 03:34:46 CST 2009
MOXA:~# hwclock
Wed 16 Dec 2009 03:35:16 AM CST -0.017600 seconds
MOXA:~# date 121616352009
Wed Dec 16 16:35:00 CST 2009
MOXA:~# hwclock -w
MOXA:~# date ; hwclock
Wed Dec 16 16:36:12 CST 2009
Wed 16 Dec 2009 03:38:13 AM CST -0.016751 seconds
MOXA:~#
```

## NTP Client

The V2101-LX has a built-in NTP (Network Time Protocol) client that is used to initialize a time request to a remote NTP server. Use **ntpdate** to update the system time.

```
#ntpdate time.stdtime.gov.tw
```

```
#hwclock -w
```

Visit <http://www.ntp.org> for more information about NTP and NTP server addresses.

```
MOXA:~# date ; hwclock
Wed Dec 16 16:36:12 CST 2009
Wed 16 Dec 2009 03:38:13 AM CST -0.016751 seconds
MOXA:~#
MOXA:~# ntpdate time.stdtime.gov.tw
16 Dec 03:49:48 ntpdate[2510]: step time server 220.130.158.52 offset 155905087.9
84256 sec
MOXA:~#
MOXA:~# hwclock -w
MOXA:~# date ; hwclock
Wed Dec 16 03:51:07 CST 2009
Wed 16 Dec 2009 03:51:07 AM CST -0.016771 seconds
MOXA:~#
```



### ATTENTION

Before using the NTP client utility, check your IP address and network settings (gateway and DNS) to make sure an Internet connection is available.

## Updating the Time Automatically

This section describes how to use a shell script to update the time automatically.

### Example shell script for updating the system time periodically

```
#!/bin/sh
ntpdate time.stdtime.gov.tw
moxa@Moxa:~# You can use the time server's ip address or domain
moxa@Moxa:~# name directly. If you use domain name, you must
moxa@Moxa:~# enable the domain client on the system by updating
moxa@Moxa:~# /etc/resolv.conf file.
hwclock -w
sleep 100
moxa@Moxa:~# Updates every 100 seconds. The min. time is 100 seconds.
moxa@Moxa:~# Change 100 to a larger number to update RTC less often.
```

Save the shell script using any file name. For example, **fixtime**.

### How to run the shell script automatically when the kernel boots up

Because the root file system is mounted in Read-only mode, we need to re-mount it using writable permission.

```
moxa@Moxa:~# mount -o remount,rw /dev/hda1 /
```

Copy the example shell script **fixtime** to directory **/etc/init.d**, and then use **chmod 755 fixtime** to change the shell script mode.

```
moxa@Moxa:~# chmod 755 fixtime
```

Next, use **vi** editor to edit the file **/etc/inittab**.

```
moxa@Moxa:~# vi /etc/inittab
```

Add the following line to the bottom of the file:

```
ntp : 2345 : respawn : /etc/init.d/fixtime
```

After you finish writing or modifying the code, remember to execute "umount /" to change the root directory back to Read-only mode.

```
moxa@Moxa:~# umount /
```

Use the command **#init q** to re-initialize the kernel.

```
moxa@Moxa:~# init q
```

## Enabling and Disabling Daemons

The following daemons are enabled when the V2101-LX boots up for the first time.

The following daemons are enabled when the V2101-LX boots up for the first time.

- **snmpd**           SNMP Agent Daemon
- **telnetd**        Telnet Server/Client Daemon
- **inetd**          Internet Daemons
- **ftpd**           FTP Server/Client Daemon
- **sshd**           Secure Shell Server Daemon
- **httpd**          Apache WWW Server Daemon

Type the command **ps -ef** to list all processes currently running.

```

MOXA:~# ps -ef
  PID  Uid      VmSize  Stat  Command
    1  root      1296  S    init
    2  root           S    [keventd]
    3  root           S    [ksoftirqd_CPU0]
    4  root           S    [kswapd]
    5  root           S    [bdflush]
    6  root           S    [kupdated]
    7  root           S    [mtdblockd]
    8  root           S    [khubd]
   10  root           S    [jffs2_gcd_mtd3]
   32  root           D    [ixp425_csr]
   38  root      1256  S    stdef
   47  root      1368  S    /usr/sbin/inetd
   53  root      4464  S    /usr/sbin/httpd
   63  nobody    4480  S    /usr/sbin/httpd
   64  nobody    4480  S    /usr/sbin/httpd
   65  nobody    4480  S    /usr/sbin/httpd
   66  nobody    4480  S    /usr/sbin/httpd
   67  nobody    4480  S    /usr/sbin/httpd
   92  bin       1460  S    /sbin/portmap
  105  root      1556  S    /usr/sbin/rpc.statd
  109  root      4044  S    /usr/sbin/snmpd -s -l /dev/null
  111  root      2832  S    /usr/sbin/snmptrapd -s
  140  root      1364  S    /sbin/cardmgr
  144  root      1756  S    /usr/sbin/rpc.nfsd
  146  root      1780  S    /usr/sbin/rpc.mountd
  153  root      2960  S    /usr/sbin/sshd
  161  root      1272  S    /bin/reportip
  162  root      3464  S    /bin/massupfirm
  163  root      1532  S    /sbin/getty 115200 ttyS0
  164  root      1532  S    /sbin/getty 115200 ttyS1
  166  root      3464  S    /bin/massupfirm
  168  root      3464  S    /bin/massupfirm
  171  root      3652  S    /usr/sbin/sshd
  172  root      2200  S    -bash
  174  root      1592  S    ps -ef
MOXA:~#

```

To run a private daemon, you can edit the file **rc.local** as follows:

1. Because the root file system is mounted in Read-only mode, you need to re-mount it with write permission.

```
MOXA:~# mount -o remount,rw /dev/hda1 /
```

2. Type `cd /etc/` to change directories.

```
MOXA:~# cd /etc/
```

3. Type `vi rc.local` to edit the configuration file with vi editor.

```
MOXA:/etc/# vi rc.local
```

4. Next, add the application daemon that you want to run. We use the example program **tcps2-release** which you can find in the CD to illustrate, and configure it to run in the background.

```

# !/bin/sh
moxa@Moxa:~# Add you want to run daemon
/root/tcps2-release &~

```

5. After you finish writing or modifying the code, remember to execute `"umount /"` to change the root directory back to Read-only mode.

```
MOXA:~# umount /
```

6. You should be able to find the enabled daemon after you reboot the system.

```
MOXA:~# ps -ef
  PID  Uid          VmSize  Stat  Command
    1  root          1296  S     init
    2  root                S     [keventd]
    3  root                S     [ksoftirqd_CPU0]
    4  root                S     [kswapd]
    5  root                S     [bdflush]
    6  root                S     [kupdated]
    7  root                S     [mtdblockd]
    8  root                S     [khubd]
   10  root                S     [jffs2_gcd_mtd3]
   32  root                D     [ixp425_csr]
   38  root          1256  S     stdef
   47  root          1368  S     /usr/sbin/inetd
   53  root          4464  S     /usr/sbin/httpd
   63  nobody       4480  S     /usr/sbin/httpd
   64  nobody       4480  S     /usr/sbin/httpd
   65  nobody       4480  S     /usr/sbin/httpd
   66  nobody       4480  S     /usr/sbin/httpd
   67  nobody       4480  S     /usr/sbin/httpd
   92  bin           1460  S     /sbin/portmap
   97  root          1264  S     /root/tcps2-release
  105  root          1556  S     /usr/sbin/rpc.statd
  109  root          4044  S     /usr/sbin/snmpd -s -l /dev/null
  111  root          2832  S     /usr/sbin/snmptrapd -s
  140  root          1364  S     /sbin/cardmgr
  144  root          1756  S     /usr/sbin/rpc.nfsd
  146  root          1780  S     /usr/sbin/rpc.mountd
  153  root          2960  S     /usr/sbin/sshd
  161  root          1272  S     /bin/reportip
  162  root          3464  S     /bin/massupfirm
  163  root          1532  S     /sbin/getty 115200 ttyS0
  164  root          1532  S     /sbin/getty 115200 ttyS1
  166  root          3464  S     /bin/massupfirm
  168  root          3464  S     /bin/massupfirm
  171  root          3652  S     /usr/sbin/sshd
  172  root          2200  S     -bash
  174  root          1592  S     ps -ef
MOXA:~#
```

## Setting the Run-Level

To set the Linux run-level and execution priority of a program, use the following command (because the root file system is mounted in Read-only mode, we need to re-mount it with write permission).

```
MOXA:~# mount -o remount,rw /dev/hda1 /
```

Edit a shell script to execute **/root/tcps2-release** and save to **tcps2** as an example.

```
#cd /etc/rc2.d
```

```
#ln -s /etc/root/tcps2 S60tcps2
```

or

```
#ln -s /etc/root/tcps2 k30tcps2
```



```
MOXA:~# cd /etc/rc2.d
MOXA:/etc/rc2.d#
MOXA:/etc/rc2.d# ls
S19nfs-common      S25nfs-user-server  S99showreadyled
S20snmpd           S55ssh
S24pcmcia          S99rmnologin
MOXA:/etc/rc2.d#
MOXA:/etc/rc2.d# ln -s /root/tcps2-release S60tcps2
MOXA:/etc/rc2.d# ls
S19nfs-common      S25nfs-user-server  S99rmnologin
S20snmpd           S55ssh              S99showreadyled
S24pcmcia          S60tcps2
MOXA:/etc/rc2.d#
```

The command **SxxRUNFILE** has the following meaning:

**S:**           **Start the run file while Linux boots up.**  
**xx:**          **A number between 00-99. The smaller number has a higher priority.**  
**RUNFILE:**    **The script file name.**

The command **KxxRUNFILE** has the following meaning:

**K:**           **Start the run file while Linux shuts down or halts.**  
**xx:**          **A number between 00-99. The smaller number has a higher priority.**  
**RUNFILE:**    **The script file name.**

To remove the daemon, remove the run file from /etc/rc2.d by using the following command:

```
MOXA:~# rm -f /etc/rc2.d/S60tcps2
```

After you finish writing or modifying the code, remember to execute "umount /" to change the root directory back to Read-only mode.

```
MOXA:~# umount /
```

## Cron—Daemon for Executing Scheduled Commands

The Cron daemon will search **/etc/crontab** for crontab files.

Cron wakes up every minute and checks each command to see if it should be run in that minute. When executing commands, output is mailed to the owner of the **crontab** (or to the user named in the MAILTO environment variable in the **crontab**, if such a user exists).

Modify the file **/etc/crontab** to set up your scheduled applications. **Crontab** files have the following format:

mm	h	dom	mon	dow	user	command
minute	hour	date	month	week	user	command
0-59	0-23	1-31	1-12	0-6 (0 is Sunday)		

For example, if you want to launch a program at 8:00 every day

```
#minute hour date month week user command
*      8    *    *    *    root  /path/to/your/program
```

The following example demonstrates how to use **Cron** to update the system time and RTC time every day at 8:00.

1. Write a shell script named fixtime.sh and save it to **/home/**.

```
#!/bin/sh
ntpdate time.stdtime.gov.tw
hwclock -w
exit 0
```

2. Change mode of **fixtime.sh**  
moxa@Moxa:~# **chmod 755 fixtime.sh**
3. Modify /etc/crontab file to run fixtime.sh at 8:00 every day.  
Add the following line to the end of crontab:  
**\* 8 \* \* \* root /home/fixtime.sh**

## Inserting a USB Storage Device into the Computer

Since mounting USB storage devices manually can be difficult, a Debian package named **usbmount** to mount the USB drivers automatically. **usbmount** relies on **udev** to mount USB storage devices automatically at certain mount points. The USB storage devices will be mounted on **/media/usb0**, **/media/usb1**, etc.

```
MOXA:~# mount
/dev/hda1 on / type ext2 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
procbususb on /proc/bus/usb type usbfs (rw)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
/dev/hdb2 on /home type ext2 (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/sda1 on /media/usb0 type vfat
(rw,noexec,nodev, sync,noatime,gid=25,dmask=0007,fmask=0117)
/dev/sdb1 on /media/usb1 type vfat
(rw,noexec,nodev, sync,noatime,gid=25,dmask=0007,fmask=0117)
MOXA:~#
```



### ATTENTION

Remember to type the command **# sync** before you disconnect the USB storage device. If you do not issue the command, you may lose data.



### ATTENTION

Remember to exit the **/media/usb0** or **/media/usb1** directory when you disconnect the USB storage device. If you stay in **/media/usb0** or **/media/usb1**, the automatic un-mount process will fail. If that happens, type **# umount /media/usb0** to un-mount the USB device manually.

## VGA & LVDS Configuration

The V2101-LX has dual display outputs: VGA and LVDS. The default text console uses VGA output; you may change to LVDS output via BIOS settings. Refer to the BIOS chapter of the hardware user's manual for details.

Note that only clone mode and single mode are supported. Generic LVDS panels come without OSD, so edit **/etc/X11/xorg.conf** and select an appropriate configuration ID for your LVDS panel before running x-window desktop. Refer to the following for configuration parameters.

Configuration ID	Panel Resolution	Manufacturer
1	640*480	General
2	800*600	General
3	1024*768	General
4	640*480	NEC
5	800*600	NEC
6	1024*768	TMD
7	1024*600	Samsung
8	1024*768	Samsung
9	1024*768	Sharp
10	1280*800	Samsung
11	1366*768	General

1. Set the "ConfigId" value for your LVDS panel resolution. (See the following example where value 3 marked in blue.)

```
MOXA:~# vi /etc/X11/xorg.conf
moxa@Moxa:~# Primary (First/only) display
Section "Device"
    Identifier "Intel_IEGD-0"
    Driver      "iegd"
    VendorName "Intel (R) DEG"
    BoardName  "Embedded Graphics"
    BusID      "0:2:0"
    Screen     0
    Option     "PcfVersion"          "1792"
    Option     "ConfigId"           "3"
    Option     "ALL/7/name"         "10x6Samsung"
    Option     "ALL/7/General/PortOrder" "24000"
    Option     "ALL/7/General/DisplayConfig" "2"
    Option     "ALL/7/General/DisplayDetect" "1"
    Option     "ALL/7/Port/2/General/name"          "SDVO"
    Option     "ALL/7/Port/2/General/EdidAvail"     "3"
    Option     "ALL/7/Port/2/General/EdidNotAvail"  "1"
    Option     "ALL/7/Port/2/General/Rotation"     "0"
```

Note that LVDS output does not support the auto-scaling feature. Therefore, be sure to follow all the steps for proper LDVS panel resolution setup.

2. In the same file, **/etc/X11/xorg.conf**, change the value to save screen resolution as the default value. In the following example we use 1024x768 as the default setting. Save the file to exit.

```
MOXA:~# mount -o remount,rw /
MOXA:~#vi /etc/X11/xorg.conf
...
Section "Screen"
    Identifier "Screen0"
    Device "Intel_IEGD-0"
    Monitor "Monitor0"
    SubSection "Display"
        Modes "1024x768"
    EndSubSection
EndSection
```

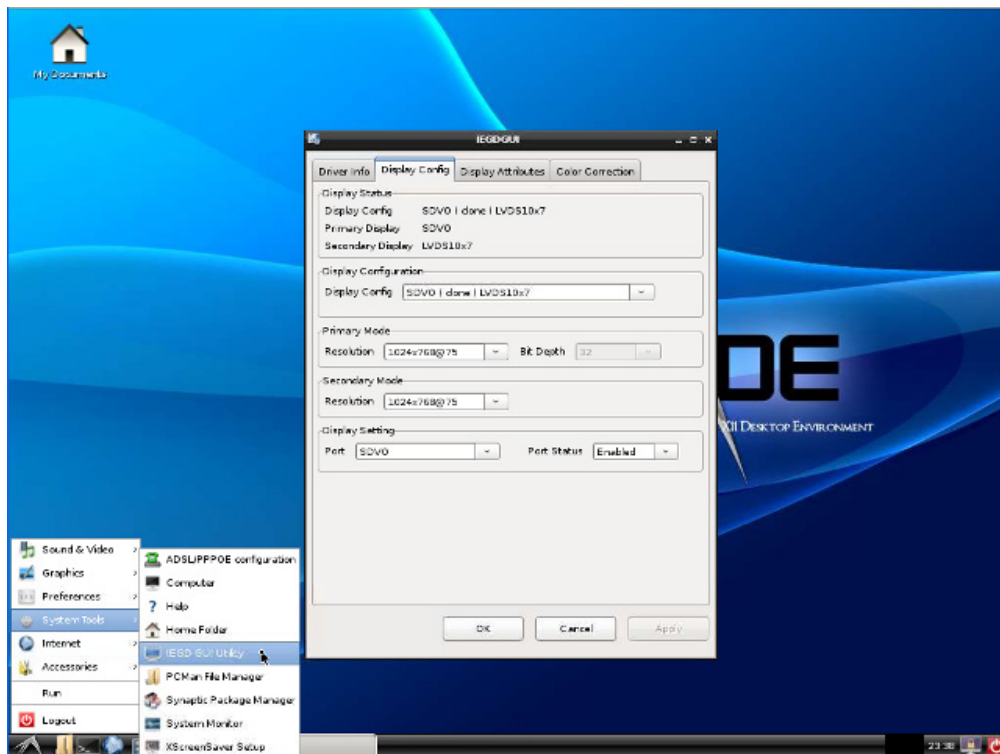
- Next, launch X-window desktop environment by default console:

```
MOXA:~# startx
```



Once the X-window desktop has been launched, use following application menu to change screen configurations:

Select **Applications** → **System Tools** → **IEGD GUI Utility**.



The V2101 IEGD driver supports three display configuration modes: **SDVO (single)**, **LVDS (single)**, **SDVO(clone)LVDS**. You may choose the display interface in this step. Note that the default setting is SDVO (clone)LVDS. In this mode, the setting of LVDS output will follow the SDVO mode.

If you would like to use X-window as the default console when the system launches, you will need to install the package **gdm**. Refer to the following steps.

1. Mount root file system as writable status.
2. Run **apt-get install gdm**.
3. When finished, reboot the computer.

```
Moxa:~# mount -o remount,rw /
Moxa:~# apt-get install gdm
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  gdm-themes
The following NEW packages will be installed:
  gdm gdm-themes
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 0B/8539kB of archives.
After this operation, 21.0MB of additional disk space will be used.
Do you want to continue [Y/n]? y
Preconfiguring packages ...
Selecting previously deselected package gdm.
(Reading database ... 53291 files and directories currently installed.)
Unpacking gdm (from ../gdm_2.20.7-4lenny1_i386.deb) ...
Selecting previously deselected package gdm-themes.
Unpacking gdm-themes (from ../gdm-themes_0.6.1_all.deb) ...
Processing triggers for man-db ...
Processing triggers for menu ...
Setting up gdm (2.20.7-4lenny1) ...
Adding group `gdm' (GID 111) ...
Done.
Warning: The home dir /var/lib/gdm you specified already exists.
Adding system user `gdm' (UID 110) ...
Adding new user `gdm' (UID 110) with group `gdm' ...
The home directory `/var/lib/gdm' already exists. Not copying from `/etc/skel'.
adduser: Warning: The home directory `/var/lib/gdm' does not belong to the user
you are currently creating.
usermod: no changes
usermod: no changes
usermod: no changes
Scheduling reload of GNOME Display Manager configuration: gdm.
Setting up gdm-themes (0.6.1) ...
Processing triggers for menu ...
MOXA:~#
MOXA:~#Reboot
```

After the system restarts, X-window will launch automatically.

If you would like to remove the package and return to text console, use the following command.

```
MOXA:~#apt-get remove gdm
```

Reboot the computer to complete.

**ATTENTION**

Due to VGA driver limitations (refer to <http://edc.intel.com/Download.aspx?id=2390&returnurl=/Software/Downloads/IEGD/default.aspx>), system response may slow or the screen may become blank or corrupted on Linux when switching between X-window and console text mode in single and clone display configurations on US15W. To remedy this, reboot the system when switching between the console text mode and X-window.

## Checking the Linux Version

The program **uname**, which stands for "Unix Name" and is part of the Unix operating system, prints the name, version, and other details about the operating system running on the computer. Use the **-a** option to generate a response similar to the one shown below:

```
MOXA:~# uname -a
Linux Moxa 2.6.26-2-686 #1 SMP Sun Jul 26 21:25:33 UTC 2009 i686 GNU/Linux
MOXA:~#
```

## APT—Installing and Removing Packages

APT is the Debian tool used to install and remove packages. Before installing a package, you need to configure the apt source file, **/etc/apt/sources.list**, which is located in the read-only partition.

1. Mount the root file system with write permission.

```
MOXA:~# mount -o remount,rw /dev/hda1 /
```

2. Next, configure the **/etc/apt/sources.list** using **vi** editor.

```
MOXA:~# vi /etc/apt/sources.list

#
moxa@Moxa:~# deb cdrom:[Debian GNU/Linux 5.0.2a _Lenny_ - Official i386 NETINST
Binary-1 20
090817-16:43]/ lenny main

#deb cdrom:[Debian GNU/Linux 5.0.2a _Lenny_ - Official i386 NETINST Binary-1 200
90817-16:43]/ lenny main

deb http://ftp.us.debian.org/debian/ lenny main
deb-src http://ftp.us.debian.org/debian/ lenny main

deb http://security.debian.org/ lenny/updates main contrib
deb-src http://security.debian.org/ lenny/updates main contrib

deb http://volatile.debian.org/debian-volatile lenny/volatile main
deb-src http://volatile.debian.org/debian-volatile lenny/volatile main
```

3. Update the source list after you configure it.

```
MOXA:~# apt-get update
MOXA:~#
```

4. Once you indicate which package you want to install (**openswan**, for example), type:

```
MOXA:~# apt-get install openswan
MOXA:~#
```

5. Use one of the following commands to remove a package:

(a) For a simple package removal:

```
MOXA:~# apt-get remove openswan
MOXA:~#
```

(b) For a complete package removal:

```
MOXA:~# apt-get remove openswan --purge
MOXA:~#
```

6. If the installation is complete, remember to unmount the root directory back to read-only mode.

```
MOXA:~# umount /
MOXA:~#
```



### ATTENTION

The APT cache space `/var/cache/apt` is located in `tmpfs`. If you need to install a huge package, link `/var/cache/apt` to USB mass storage or mount it to an NFS space to generate more free space. Use `df -h` to check how much free space is available on `tmpfs`.

```
MOXA:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          1.6G  972M  560M  64% /
udev            10M   700K   9.4M   7% /dev
/dev/hda1       1.6G  972M  560M  64% /
tmpfs           502M     0   502M   0% /lib/init/rw
tmpfs           502M     0   502M   0% /dev/shm
none            502M   19M   483M   4% /tmp
/dev/hda2       199M  125M   63M  67% /home
MOXA:~#
```



### ATTENTION

You can free up the cache space with the command `# apt-get clean`

```
MOXA:~# apt-get clean
MOXA:~#
```

# Managing Communications

---

The V2101-LX ready-to-run embedded computer is a network-centric platform designed to serve as a front-end for data acquisition and industrial control applications. This chapter describes how to configure the various communication functions supported by the Linux operating system.

The following topics are covered in this chapter:

## ❑ **Configuring Network Interfaces**

- Configuring a Persistent Network Interface Naming Order
- Ethernet Interface Configuration
- Static IP Address
- Dynamic IP Address using DHCP
- Adjusting IP Addresses with ifconfig

## ❑ **Point-to-Point Over Ethernet (PPPoE) Config**

- The Easy Way: pppoeconf
- The Difficult Way (Manually)

## ❑ **Configuring a Point-to-Point Connection**

- Connecting to a PPP Server over a Hardwired Link
- Checking the Connection
- Setting up a Machine for Incoming PPP Connections

## ❑ **Serial Port Operation Mode**

## ❑ **Telnet/FTP/TFTP Server**

- Enabling a Telnet, FTP, or TFTP Server
- Disabling a Telnet/FTP/TFTP Server

## ❑ **DNS Utilities**

- Configuring the OS Hostname
- Configuring the DNS Resolver
- Configuring the Name Service Switcher

## ❑ **Apache Web Server**

- Default Homepage
- Configuring the Common Gateway Interface (CGI)
- Saving Web Pages to a USB Storage Device

## ❑ **Netfilter/iptables**

- IP Tables and IP Chains
- Understanding Rule Chains
- Understanding Basic Traffic Flows
- Connection Tracking

## ❑ **Building the Firewall**

- Policies: Setting Default Firewall Behavior
- Viewing and Manipulating Rulesets
- Writing Rulechains
- Setting Up NAT (Network Address Translation)
- Saving the Firewall

## ❑ **Setting up a Networked File System: NFS**

## ❑ **Setting Up a VPN**

## ❑ **SNMP (Simple Network Management Protocol)**



# Configuring Network Interfaces

## Configuring a Persistent Network Interface Naming Order

Debian Linux systems use the **udev** daemon to detect and enable new network interfaces and to manage the device files that are created for them. Udev must be configured with rules that enforce a **persistent interface naming order**. A persistent network interface naming order allows devices to be consistently named with the same device node every time the machine is rebooted. This is important because settings are configured with reference to a device name (e.g, eth1) associated with a particular device (e.g., your Broadcom gigabit Ethernet card). If every time the system is rebooted the system randomly rearranges the naming of your cards—for instance, assigning your gigabit Ethernet card to eth2 and your 10/100 Ethernet card to eth1—then there will be no way to maintain a consistent configuration across restarts.

The rule for setting up network interfaces with a persistent naming order is found here:

```
/lib/udev/rules.d/75-persistent-net-generator.rules
```

and it looks like this:

```
# PCI device 0x10ec:/sys/devices/pci0000:00/0000:00:1c.1/0000:02:00.0 (r8169)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:90:e8:00:de:a9", ATTR{dev_id}=="0x0", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth1"

#PCI device 0x10ec:/sys/devices/pci0000:00/0000:00:1c.0/0000:01:00.0 (r8169)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:90:e8:00:de:a8", ATTR{dev_id}=="0x0", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth0"
```

The above example indicates that the system has detected two Ethernet interfaces, and assigned them the names eth0 (which is associated with the MAC address 00:90:e8:00:de:a8) and eth1 (associated with the MAC address 00:90:e8:00:de:a9).



### ATTENTION

When replacing or connecting a network interface, the system may fail to remove the old record from `/etc/udev/rules.d/70-persistent-net.rules`. This could cause network interfaces to be detected abnormally. To avoid this problem, simply delete the **70-persistent-net.rules** file and reboot the system.



### ATTENTION

It may also be necessary to configure a persistent naming order for other system peripherals (e.g., storage drives); to find out more, you may start with the Writing Udev Rules tutorial, found at Ractivated.Net:

[http://www.reactivated.net/writing\\_udev\\_rules.html](http://www.reactivated.net/writing_udev_rules.html)

Symantec also offers an effective tutorial, **Setting Persistent SCSI Device Names On Linux Using UDEV**, found here:

<http://www.symantec.com/business/support/index?page=content&id=TECH71007>

To get an idea of what Udev can do for you, check out this Linux For You article from 2012, **Some Nifty udev Rules and Examples**:

<http://www.linuxforu.com/2012/06/some-nifty-udev-rules-and-examples/>

## Ethernet Interface Configuration

The V2101-LX computer has two 10/100/1000 Ethernet ports named LAN1 and LAN2. The default IP addresses and netmasks of these network interfaces are:

	Default IP Address	Netmask
LAN1	192.168.3.127	255.255.255.0
LAN2	192.168.4.127	255.255.255.0

These network settings can be modified by changing the **interfaces** (/etc/networking/interfaces) configuration file, or they can be adjusted temporarily with the **ifconfig** command.

The file used for configuring network interfaces is the **networking interfaces configuration** file, located in the /etc/network directory. The /etc/network/interfaces file is where you will configure Ethernet LAN ports for either static or dynamic (DHCP) IP addressing. To edit this file directly, open the network configuration file with your preferred editor (below, we use VI):

```
moxa@MOXA:~#/etc/network# sudo vi interfaces
```

## Static IP Address

The default static IP addresses can be modified. Below, we show the default configuration; changing these values will change the addressing and broadcast parameters used by the associated interface.

```
### The loopback network interface
auto lo
iface lo inet loopback
### The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.3.127
    netmask 255.255.255.0
    broadcast 192.168.3.255
auto eth1
iface eth1 inet static
    address 192.168.4.127
    netmask 255.255.255.0
    broadcast 192.168.4.255
```

## Dynamic IP Address using DHCP

To configure one or both LAN ports to receive an IP address through dynamic assignment, replace **static** with **dhcp** and then comment out the rest of the lines. The eth0 interface is shown below, as an example.

```
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet dhcp
#     address 192.168.3.127
#     netmask 255.255.255.0
#     broadcast 192.168.3.255
```

After modifying the boot settings of the LAN interface, issue the following command to immediately activate the new LAN settings:

```
moxa@MOXA:~# sudo service networking restart
```

## Adjusting IP Addresses with ifconfig

IP settings can be adjusted during run-time, but the new settings will not be saved to the flash ROM without modifying the file `/etc/network/interfaces`. For example, the following command changes the IP address of **LAN1** to **192.168.1.1**.

```
moxa@MOXA:~# sudo ifconfig eth0 192.168.1.1
```

## Point-to-Point Over Ethernet (PPPoE) Config

### The Easy Way: pppoeconf

The easiest way to set up a PPPoE connection is to install the Debian package, `pppoeconf`. This is a script that automates the PPPoE configuration process; it may be used on any connection that is directly linking to an ADSL or other PPPoE modem.

Use `apt-get` or `Aptitude` to install `pppoeconf`:

```
moxa@Moxa: ~# apt-get pppoeconf
```

After installing `pppoeconf`, call it from the command line:

```
moxa@MOXA:~# pppoeconf
```

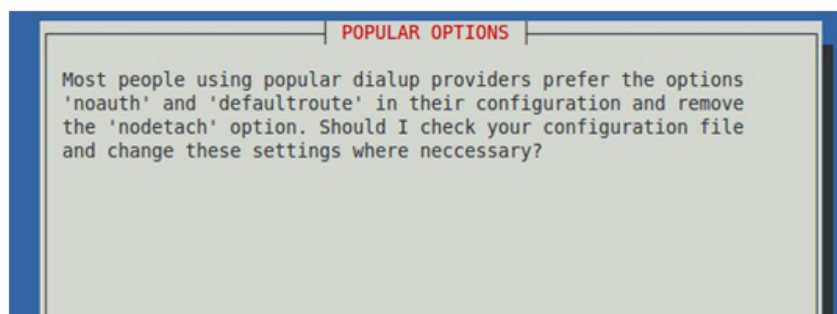
Next, a dialog will appear telling you `pppoeconf` is locating your "access concentrator." If your DSL or ADSL modem is connected to an active LAN interface, `pppoeconf` will find it.

If there are no available concentrators, `pppoeconf` will tell you, and exit; if this happens, check to see you're your modems are connected properly.

If `pppoeconf` successfully discovers a concentrator on an available interfaces, it will return this screen:

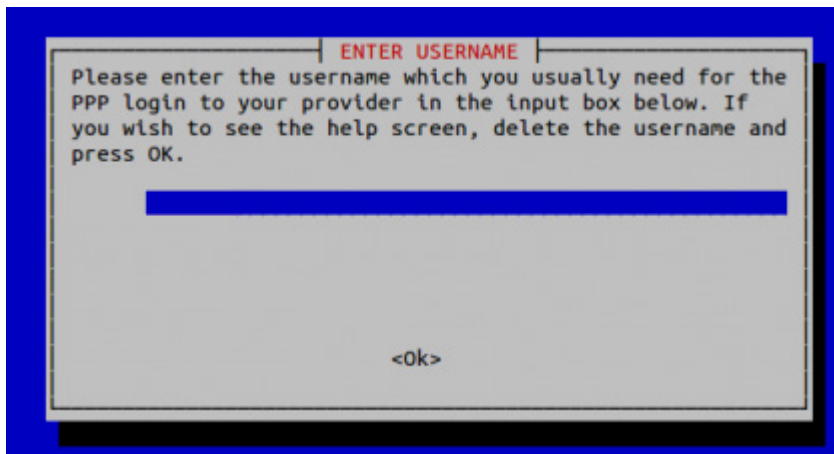


Answer yes. You will then see this screen:



Noauth indicates that the peer does not need to authenticate itself. Nodetach indicates that the connection will not detach from the controlling terminal. Without this option, if a serial device other than the terminal on the standard input is specified, pppd will fork to become a background process.

After choosing whether or not to use noauth and nodetach, the pppoeconf will next ask you for your username and password.



Next, enter your password:



Finally, you will need to choose whether or not your PPPoE provider will supply you with DNS server addresses. These addresses are necessary for DNS resolution (see below, in the final step of the next section, **Setting up DNS**). It is preferable to click **Yes**, here; however, if your PPPoE provider does not supply these addresses automatically, click **No** and remember that you will need to enter the DNS server addresses into `/etc/resolv.conf` by hand.



## The Difficult Way (Manually)

You may wish or need to connect to your PPPoE provider by manually configuring a connection. Here is how. Use the following procedure to configure PPPoE:

1. Connect the V2101-LX's LAN port to an ADSL modem (you may use a cable, HUB, or switch).
2. Log in to the V2101-LX as the root user.
3. Edit the file `/etc/ppp/pap-secrets` and add the following entry in the place indicated below:

```

"username@YourProvider.net" * "password" *

# ATTENTION: The definitions here can allow users to login without a
# password if you don't use the login option of pppd! The mgetty Debian
# package already provides this option; make sure you don't change that.

# INBOUND connections

# Every regular user can use PPP and has to use passwords from /etc/passwd
*      hostname      ""      *
"username@YourProvider.net" *      "password"      *

# UserIDs that cannot use PPP at all. Check your /etc/passwd and add any
# other accounts that should not be able to use pppd!
guest  hostname      "*"      -
master hostname      "*"      -
root   hostname      "*"      -
support hostname     "*"      -
stats  hostname      "*"      -

# OUTBOUND connections

```

**username@YourProvider.net** is the username obtained from the ISP to log in to the ISP account.  
**password** is the corresponding password for the account.

4. Edit the file `/etc/ppp/options` and add **plugin rp-pppoe** in the indicated place:

```

# Wait for up n milliseconds after the connect script finishes for a valid
# PPP packet from the peer. At the end of this time, or when a valid PPP
# packet is received from the peer, pppd will commence negotiation by
# sending its first LCP packet. The default value is 1000 (1 second).
# This wait period only applies if the connect or pty option is used.
#connect-delay <n>

# Load the pppoe plugin
plugin rp-pppoe.so

# ---<End of File>---

```

5. If you connecting over LAN1, use the template below to create a file `/etc/ppp/options.eth0`. LAN2 should be named `/etc/ppp/options.eth1`. All interfaces follow this convention.

```

name username@YourProvider.net
mtu 1492
mru 1492
defaultroute
noipdefault
~
~
"/etc/ppp/options.eth0" 5 lines, 67 characters

```

Type your username (the one you set in the `/etc/ppp/pap-secrets` and `/etc/ppp/chap-secrets` files) after the **name** option. You may add other options as needed.

6. **Set up DNS:** If you are using DNS servers supplied by your ISP, edit the file `/etc/resolv.conf` by adding the following lines of code:

```
nameserver ip_addr_of_first_dns_server
nameserver ip_addr_of_second_dns_server
```

For example:

```
nameserver 168.95.1.1
nameserver 139.175.10.20
```

```
moxa@MOXA:~# cat /etc/resolv.conf
#
# resolv.conf This file is the resolver configuration file
# See resolver(5).
#
nameserver 168.95.1.1
nameserver 139.175.10.20
#/etc#
```

Now, you should be able to use the following command to establish a **pppoe** connection:

```
moxa@Moxa:~# pppd eth0
```

If you want to disconnect the connection, you may use the kill command to kill the **pppd** process.

```
moxa@Moxa:~# kill -9 pppd
```

### Notes:

1. If the ADSL modem is connected to the **LAN1** port, the connection will be named **eth0**. If the ADSL modem is connected to **LAN2**, it should be named **eth1**, etc.
2. Type `moxa@Moxa: ~# ifconfig ppp0` to check if the connection is OK. If the connection is OK, you should see the IP address of ppp0. You may use the **ping** command to test the IP address.

```
ppp0      Link encap Point-to-Point Protocol
          inet addr 192.76.32.3  P-t-P 129.67.1.165 Mask 255.255.255.0
          UP POINTOPOINT RUNNING  MTU 1500  Metric 1
          RX packets 33 errors 0 dropped 0 overrun 0
          TX packets 42 errors 0 dropped 0 overrun 0
```

## Configuring a Point-to-Point Connection

PPP (Point to Point Protocol) is used to run IP (Internet Protocol) and other network protocols over a serial link. PPP can be used for direct serial connections (using a null-modem cable) over a Telnet link, and links established using a modem over a telephone line.

Modem/PPP access is almost identical to connecting directly to a network through the V2101-LX Ethernet port. Since PPP is a peer-to-peer system, the V2101-LX can also use PPP to link two networks (or a local network to the Internet) to create a Wide Area Network (WAN).



### ATTENTION

The following links will give you more information about setting up PPP:

<http://tldp.org/HOWTO/PPP-HOWTO/index.html>  
<http://axion.physics.ubc.ca/ppp-linux.html>

The following is an AT command used to connect to a PPP server by modem. Use this command for old ppp servers that prompt for a login name (replace **username** with the correct name) and password (replace

**password** with the correct password). Note that **debug crtscts** and **defaultroute 192.XXX.XX.XXX** are optional.

```
moxa@Moxa:~# pppd connect `chat -v "" ATDT5551212 CONNECT ""` login: username \
password: password' /dev/ttyM0 115200 \
debug crtscts modem defaultroute 192.1.1.17
```

If the PPP server does not prompt for the username and password, the command should be entered as follows (replace "username" with the correct username and replace "password" with the correct password):

```
moxa@Moxa:~# pppd connect `chat -v "" ATDT5551212 CONNECT ""` user username
password password /dev/ttyM0 115200 crtscts modem
```

The pppd options are described below:

**connect `chat etc...`** This option gives the command to contact the PPP server. The **chat** program is used to dial a remote computer. The entire command is enclosed in single quotes because pppd expects a one-word argument for the **connect** option. The options for **chat** are given below:

**-v** verbose mode; log what we do to syslog

**""** Double quotes—don't wait for a prompt, but instead do ... (note that you must include a space after the second quotation mark)

**ATDT5551212** Dial the modem, and then ...

**CONNECT** Wait for an answer.

**""** Send a return (null text followed by the usual return)

**login: username word: password**  
Log in with username and password.

Note: Refer to the chat man page, chat.8, for more information about the **chat** utility.

**/dev/** Specify the callout serial port.

**115200** The baud rate.

**debug** Log status in syslog.

**crtscts** Use hardware flow control between the computer and modem (at baudrate of 115200 this is a must).

**modem** Indicates that this is a modem device; pppd will hang up the phone before and after making the call.

**defaultroute** Once the PPP link is established, make it the default route; if you have a PPP link to the Internet, this is probably what you want.

**192.1.1.17** This is a degenerate case of a general option of the form x.x.x.x:y.y.y.y. Here x.x.x.x is the local IP address and y.y.y.y is the IP address of the remote end of the PPP connection. If this option is not specified, or if just one side is specified, then x.x.x.x defaults to the IP address associated with the local machine's hostname (located in **/etc/hosts**), and y.y.y.y is determined by the remote machine.

## Connecting to a PPP Server over a Hardwired Link

If a username and password are not required, use the following command (note that **noipdefault** is optional):

```
moxa@Moxa:~# pppd connect `chat -v" "" ""` noipdefault /dev/ttyM0 19200 crtscts
```

If a username and password are required, use the following command (note that **noipdefault** is optional, and the username and password are both "root"):

```
moxa@Moxa:~# pppd connect `chat -v" "" ""` user root password root \
noipdefault /dev/ttyM0 19200 crtscts
```

## Checking the Connection

Once you have set up a PPP connection, there are some steps you can take to test the connection. First, type:

```
moxa@Moxa:~# ifconfig
```

After executing the command, you should be able to see all of the available network interfaces.

**ppp0** should be one of the network interfaces. You should recognize the first IP address as the IP address of the computer, and the **P-t-P address** is the address of the server. The output should be similar to this:

```
lo      Link encap Local Loopback
        inet addr 127.0.0.1  Bcast 127.255.255.255 Mask 255.0.0.0
        UP LOOPBACK RUNNING  MTU 2000  Metric 1
        RX packets 0 errors 0 dropped 0 overrun 0

ppp0    Link encap Point-to-Point Protocol
        inet addr 192.76.32.3  P-t-P 129.67.1.165 Mask 255.255.255.0
        UP POINTOPOINT RUNNING  MTU 1500  Metric 1
        RX packets 33 errors 0 dropped 0 overrun 0
        TX packets 42 errors 0 dropped 0 overrun 0
```

Now, type:

```
moxa@Moxa:~# ping XXX.XX.XXX.XXX
```

where **XXX.XX.XXX.XXX** is the address of your name server. The output should be similar to the following:

```
moxa@MOXA:~# sudo ping 129.67.1.165
PING 129.67.1.165 (129.67.1.165): 56 data bytes
64 bytes from 129.67.1.165: icmp_seq=0 ttl=225 time=268 ms
64 bytes from 129.67.1.165: icmp_seq=1 ttl=225 time=247 ms
64 bytes from 129.67.1.165: icmp_seq=2 ttl=225 time=266 ms
^C
--- 129.67.1.165 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 247/260/268 ms
moxa@MOXA:~#
```

Try typing:

```
moxa@Moxa:~# netstat -nr
```

You should see three routes similar to the following:

```
Kernel routing table
Destination Gateway Genmask Flags Metric Ref Use
iface
129.67.1.165 0.0.0.0 255.255.255.255 UH 0 0 6
ppp0
127.0.0.0 0.0.0.0 255.0.0.0 U 0 0 0 lo
0.0.0.0 129.67.1.165 0.0.0.0 UG 0 0 6298
Ppp0
```

If your output looks similar but does not have the “destination 0.0.0.0” line (which refers to the default route used for connections), you may have run `pppd` without the **defaultroute** option. At this point, you can try using Telnet, ftp, or finger, bearing in mind that you will have to use numeric IP addresses unless you have configured `/etc/resolv.conf` correctly.



## Setting up a Machine for Incoming PPP Connections

### Method 1: pppd dial-in with pppd commands

This first example applies to using a modem, and requiring authorization with a username and password.

```
#pppd /dev/ttyM0 115200 crtscts modem 192.168.16.1:192.168.16.2 login auth
```

You should also add the following line to the file `/etc/ppp/pap-secrets`:

```
* * "" *
```

The first star (\*) lets everyone login. The second star (\*) lets every host connect. The pair of double quotation marks (") indicates that the file `/etc/passwd` can be used to check the password. The last star (\*) is to let any IP connect.

The following example does not check the username and password:

```
moxa@Moxa:~# pppd/dev/ttyM0 115200 crtscts modem 192.168.16.1:192.168.16.2
```

### Method 2: pppd dial-in with pppd script

Configure a dial-in script `/etc/ppp/peer/dialin`

```
# You usually need this if there is no PAP authentication
noauth
#auth
#login

# The chat script (be sure to edit that file, too!)
init "/usr/sbin/chat -v -f /etc/ppp/ppp-ttyM0.chat"

# Set up routing to go through this PPP link
defaultroute

# Default modem (you better replace this with /dev/ttySx!)
/dev/ttyM0

# Speed
115200

# Keep modem up even if connection fails
persist
crtscts
modem
192.168.16.1:192.168.16.2
debug
-detach
```

Configure the chat script `/etc/ppp/ppp-ttyM0.chat`

```
SAY 'Auto Answer ON\n'
'' ATs0=1
```

Start the **pppd** dial-in service.

```
moxa@MOXA:~# sudo pppd call dialin
```

**ATTENTION**

If you would like to have auto dial-in service, you can launch the dial-in service in `/etc/inittab` with the respawn command:

```
moxa@MOXA:~# sudo echo "p0:2345:respawn:pppd call dialin" >> /etc/inittab
```

## Serial Port Operation Mode

The V2101 computer has 2 serial ports named COM1 and COM2. The ports support RS-232, RS-422, 2-wire RS-485, and 4-wire RS-485 operation modes with baudrate settings up to 921600 bps.

By default, the serial interface is set to RS-232. You can use the **setinterface** command to change the serial port operation mode, as indicated below:

**setinterface device-node [interface-no]**

device-node: /dev/ttyMn; n = 0,1,2,...  
interface-no: [see following table]:

interface-no	Operation Mode
None	Display current setting
0	RS-232
1	2-wire RS-485
2	RS-422
3	4-wire RS-485

For example, use the following commands to set `/dev/ttyM0` to RS-422:

```
OXA:/ dev# setinterface /dev/ttyS0
Usage: setinterface device-node [interface-no]
      device-node    - /dev/ttyS0 ~ /dev/ttyS1
      interface-no   - following:
      0 - set to RS232 interface
      1 - set to RS485-2 WIRES interface
      2 - set to RS422 interface
      3 - set to RS885-4 WIRES interface
MOXA:/ dev# setinterface /dev/ttyS0 0
MOXA:/ dev# setinterface /dev/ttyS0 2
```

## Telnet/FTP/TFTP Server

For security reasons, the V2101-LX only supports SSH and SFTP. The Telenet, FTP, and TFTP are installed, but have been disabled. Moxa strongly recommends against the use of Telnet or FTP, both of which are considered deprecated, today. However, if you wish to use one of these services, you may follow the directions below to enable or disable these services.

### Enabling a Telnet, FTP, or TFTP Server

The following example shows the default content of the file `/etc/inetd.conf`. For security's sake, the Telnet, FTP, and TFTP servers are disabled by default. To enable these services, add the following content to `/etc/inetd.conf`:

```
telnet  stream  tcp  nowait  telnetd  /usr/sbin/tcpd  /usr/sbin/in.telnetd
ftp     stream  tcp  nowait  root    /usr/sbin/tcpd  /usr/sbin/proftpd
```

```
...
tftp    dgram    udp wait    nobody    /usr/sbin/tcpd    /usr/sbin/in.tftpd    /srv/tftp
```

Then restart the inetd service:

```
moxa@MOXA:~# sudo service openbsd-inetd restart
```

## Disabling a Telnet/FTP/TFTP Server

If, after enabling one of these servers, you wish to disable it again you may do so by commenting out the relevant line inserting a hash (#) as the line's first character. Below, the **TFTP** server has been disabled using this method.

```
telnet  stream  tcp  nowait  telnetd  /usr/sbin/tcpd  /usr/sbin/in.telnetd
ftp     stream  tcp  nowait  root     /usr/sbin/tcpd  /usr/sbin/proftpd
...
#tftp   dgram   udp  wait    nobody   /usr/sbin/tcpd  /usr/sbin/in.tftpd  /srv/tftp
```

As with any other changes to the inet.d configuration, you must restart the inetd service for the changes to take effect.

```
moxa@MOXA:~# sudo service openbsd-inetd restart
```

## DNS Utilities

Basic DNS utilities are responsible for managing a system's hostname, DNS resolver, and the name service switch. The three configuration files associated with these services are `/etc/hostname`, `/etc/resolv.conf`, and `/etc/nsswitch.conf`.

## Configuring the OS Hostname

When remotely administrating large networks, it is desirable to provide each computer with a descriptive hostname. This is set by changing the **hostname** file; `/etc/hostname` is a file with a single line that contains the hostname, which can only contain the ascii characters a through z, the numbers 0 through 9, and a hyphen. Hostnames must not include dots (periods), because the hostname is used as part of a fully qualified URL.

1. To change the hostname, use the following command:

```
moxa@MOXA:~# sudo echo "your-preferred-hostname" > /etc/hostname
```

2. Load the new hostname:

```
moxa@MOXA:~# sudo /etc/init.d/hostname.sh start
```

3. Check the new hostname.

```
moxa@MOXA:~# hostname
your-preferred-hostname
```

## Configuring the DNS Resolver

This is the file most in need of updating when configuring DNS. For example, before using the command

```
moxa@Moxa:~# ntpdate time.stdtime.gov.tw
```

to update the system time, you will need to add a DNS server address to the resolver configuration. Ask your network administrator for addresses to preferred DNS servers. Each server's address is specified by prefacing the line with **nameserver**. For example, to add a DNS server with IP address is 168.95.1.1 to `/etc/resolv.conf`, you would simply append **nameserver 168.95.1.1** to the end of the file.

```

moxa@MOXA:~#/etc# echo "nameserver 168.95.1.1" >> resolv.conf
moxa@MOXA:~#/etc# cat resolv.conf
# resolv.conf This file is the resolver configuration file
# See resolver(5).
#
#nameserver 192.168.1.16
nameserver 140.115.1.31
nameserver 140.115.236.10
nameserver 168.95.1.1

```

## Configuring the Name Service Switcher

The name service switcher configuration file is **nsswitch.conf**; this file defines in what sequence system databases will be referenced to retrieve name service information when resolving URLs to IP addresses. The file is plain ASCII text, with columns separated by spaces or tab characters. The first column specifies the database name. The remaining columns describe the order of sources to query and a limited set of actions that can be performed by lookup result; the sources will be referenced in the order they appear on the line, from right to left.

Five service specifications may be indicated for any source: **files**, **db**, **nis**, **nisplus**, or **compat**. For the hosts database, you may also specify **dns**; compatibility mode (**compat**) may only be used with the **passwd**, **group**, and **shadow** databases. Use of the **files** source will have the name service switcher search the /etc directory to find a file that matches the source name (e.g., /etc/hosts, /etc/passwd, /etc/group), and then that file will be used. By omitting **dns** or **files** you may effectively disable **dns** or the local hosts file for URL resolution.

```

# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:          compat
group:           compat
shadow:         compat

hosts:           files dns
networks:        files

protocols:       db files
services:        db files
ethers:          db files
rpc:             db files

netgroup:        nis

```

## Apache Web Server

The Apache config directory houses four basic directories: sites-enabled, mods-enabled, sites-available, and mods-available. The **sites-enabled** directory is where active websites are enabled; this is done by creating a symlink into the sites-available directory. **Sites-available** is a repository for all sites, whether inactive or active. The **mods-available** directory houses Apache **software modules**, which allow administrators to adjust the size and features of the Apache webserver to the particular needs of the application. The **mods-enabled** directory enables modules to be loaded by, again, symlinking back to the relevant module located in the mods-available directory.

**ATTENTION**

There are many Apache modules that may be of use to administrators in need of customizations to their webserver, such as speeding up CGI, or building heightened security. Webserver modules and features are beyond the scope of this manual. If you wish to find a complete list and full documentation for the native modules, please refer to the Apache webserver documentation, found here:

<http://httpd.apache.org/modules/>

For a more completely list of available modules that includes third-party modules, you may refer to Wikipedia:

[http://en.wikipedia.org/wiki/List\\_of\\_Apache\\_modules](http://en.wikipedia.org/wiki/List_of_Apache_modules)

## Default Homepage

The Apache web server's main configuration file is `/etc/apache2/sites-enabled/000-default`, with the default homepage located at `/var/www/index.html`.

Before you modify the homepage, use a browser (such as Microsoft Internet Explore or Mozilla Firefox) from your PC to test if the Apache web server is working. Type the LAN1 IP address in the browser's address box to open the homepage. If the default address hasn't changed, then when you type <http://192.168.3.127/> in the address bar of your web browser you should see Apache's default web page.

## Configuring the Common Gateway Interface (CGI)

### Setting Up CGI

CGI comes already enabled. The root CGI directory (where you should put CGI scripts) is `/usr/lib/cgi-bin`. You may change this to `/var/www/cgi-bin`, if you so desire.

**ATTENTION**

If you have more questions about setting up CGI on Apache 2.2, you may refer to this web page:

<http://httpd.apache.org/docs/2.2/howto/cgi.html>

### Disabling CGI

Support for CGI scripting is enabled by default. To disable it, follow the steps below.

1. Open the configuration file for editing (below, we use VI):

```
moxa@MOXA:~# vi /etc/apache2/sites-enabled/000-default
```

Then, comment out the following lines:

```
moxa@MOXA:~# /etc# vi /etc/apache2/sites-enabled/000-default
#ScriptAlias /cgi-bin/ /usr/lib/w3m/cgi-bin/
#<Directory "/usr/lib/w3m/cgi-bin/">
#   AllowOverride None
#   Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
#   #Order allow,deny
#   Order deny,allow
#   Allow from all
#</Directory>
```

2. Re-start the apache server.

```
moxa@MOXA:~# sudo service apache2 restart
```

**ATTENTION**

If you have CGI scripts you wish to transfer to the server, make sure you make the files executable. The command for this is the **change mode** command, **chmod**. To make a file read-only but executable, you may use the numerical combination **555**. To make a file read only but available for editing by root, use the numerical key **755**. The syntax is as follows:

```
MOXA:~#chmod 555 /usr/lib/cgi-bin/[NAME OF YOUR FILE HERE]
```

## Saving Web Pages to a USB Storage Device

Some applications may have web pages that take up a lot of storage space. This section describes how to save web pages to the USB mass storage device, and then configure the Apache web server's DocumentRoot to open these pages. The files used in this example can be downloaded from the Internet.

1. Connect the USB storage device to a USB port, and check where the device is mounted:

```
moxa@Moxa:~# sudo mount
```

2. Prepare the web pages and then save the entire `/var/www` directory to the appropriate USB storage device. Normally, this should be `/media/usb0`.

```
moxa@Moxa:~# sudo cp -a /var/www/ media/usb0/
```

3. Now change the Document Root setting. Open the basic Apache config file in an editor:

```
moxa@MOXA:~# /etc# sudo vi /etc/apache2/sites-available/default
```

4. To enable Apache to read your website from the USB device, you must change the **DocumentRoot** entry in the Apache configuration file so that it points to the USB storage device. Navigate to the section beginning with DocumentRoot, and change the directory that immediately follows to **/media/usb0/www**. For a standard, unsecured html page, edit `/etc/apache2/sites-available/default` as below.

```
DocumentRoot /media/usb0/www
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
```

5. If you have CGI scripts, you must now also change the same file so that the CGI entries point to the files on the USB device. Change your basic Apache configuration file so that it matches the lines shown in red, below:

```
ScriptAlias /cgi-bin/ /media/usb0/www/cgi-bin/
<Directory "/media/usb0/www/cgi-bin/">
    AllowOverride None
    Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>
```

6. For webpages that will be connecting using the secure sockets layer, you will need to edit the SSL configuration file. Open the config file using the following command:

```
moxa@MOXA:~# /etc# sudo vi /etc/apache2/sites-available/default-ssl
```

7. Make the changes to your config file so that it matches the lines shown in red below:

```
<VirtualHost *:443>
...
    DocumentRoot /media/usb0/www
    <Directory />
        Options FollowSymLinks
```

```
        AllowOverride None
    </Directory>
    ...
    ScriptAlias /cgi-bin/ /media/usb0/www/cgi-bin/
    <Directory "/media/usb0/www/cgi-bin/">
        AllowOverride None
        Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
    ...
</VirtualHost>
```

8. Use the following compound command to restart the Apache web server:

```
MOXA:~# cd /etc/init.d && apache2 restart
```

9. Start your browser and connect to the V2101-LX by typing the current LAN1 IP address in the browser's address box.



### ATTENTION

Visit the Apache website at <http://httpd.apache.org/docs/> for more information about setting up Apache servers.

If you would like to check your website for HTML compliance, click on the following link to download the web page test suite from the World Wide Web Consortium:

<http://www.w3.org/MarkUp/Test/HTML401.zip>

# Netfilter/iptables

Netfilter is an administrative tool for setting up, maintaining, and inspecting the Linux kernel's packet filtering rule tables. Netfilter is a **stateful firewall**, which means that it filters packets by tracking connections, rather than each and every individual packet. For more information on connection tracking, see the section **Connection Tracking**, in this same chapter, below.

In Netfilter, a few fundamental **rule tables** are pre-defined, with each table containing built-in chains and user-defined chains. Tables form the highest layer of organization for Netfilter's rule sets, and **rule chains** form the middle layer, by which individual rules are ordered. Each chain is a list of rules that are applied (or not) to a packets as they traverse the chains. Each rule specifies what to do with a matching packet. A rule (such as a jump to a user-defined chain in the same table, or an order to drop a certain type of packet) is also called a **target**.

Netfilter is based around three fundamental tables: **Filter** tables, **NAT** tables, and **Mangle** tables. These tables in turn are structured around a few basic, built-in rule chains. There are five basic rule chains: PREROUTING, INPUT, FORWARDING, OUTPUT, and POSTROUTING. In addition to these five built-in chains, it is possible for users to add user-defined chains of their own devising, and insert them into the filtering and mangling procedures wherever they are needed. Thus, Netfilter may be said to have three layers: the most basic is the rules layer, the next is the chains layer (which order the rules), and the final is the table layer, which orders the rule chains.

## Overview of Basic Netfilter Architecture:

### IP Tables Review

#### The NAT Table

#### The Filter Table

#### The Mangle Table

### Understanding Rule Chains

#### The Five Built-In Rule Chains

#### User-Defined Chains

### Understanding Basic Traffic Flows

#### Netfilter Hierarchy for Incoming Packets

### Connection Tracking

## Building the Firewall: Setting Policies and Writing Rules

### Policies: Setting Default Firewall Behavior

### Viewing and Manipulating Rulesets

### Writing Rulechains

### Setting Up NAT (Network Address Translation)

### Saving the Firewall



## ATTENTION

For more information on configuring Netfilter/iptables, you may consult the official project website.

Homepage: <http://www.netfilter.org/>

Documentation: <http://www.netfilter.org/documentation/index.html#documentation-howto>

Netfilter Extensions: <http://www.netfilter.org/documentation/HOWTO//netfilter-extensions-HOWTO.html>

## IP Tables and IP Chains

The highest layer of organization in Netfilter is the **table layer**. This is where all of the **rule chains** are organized. Rule chains are ordered lists of packet filtering and packet mangling rules; each chain represents a basic flow of operations to be performed on a packet at that stage. Where chains are prioritized lists of rules, tables are prioritized lists of chains. Additionally, each of Netfilter's built-in tables comes with a set of built-in chains that are associated with it; these chains set the basic path packets will traverse as they are processed



by Netfilter. To view and manipulate (delete, flush, and add) rule tables, rulechains, and individual rules, refer to the section below, **Manipulating Rulesets**.

## The NAT Table

The NAT table is the first table that all packets will encounter; no filtering takes place in this table. The only packet alterations enforced by the NAT table are changes to the **source** and **destination** addresses; moreover, only the first packet of a new connection will traverse this table: after the first packet in a **connection** has been processed, the result will be automatically applied to all future packets in the same connection (for more information on connections, see the section **Connection Tracking**, in this same chapter, below).

When the NAT table alters the destination address (on inbound packets, in the PREROUTING chain), it is called **Destination Network Address Translation (DNAT)**, or **Port Forwarding**. When the NAT table alters the source address (on outbound packets, in the POSTROUTING chain), it is called **Source Network Address Translation (SNAT)**, or **IP Masquerading**. Netfilter conventions distinguish Masquerading from SNAT in the following way:

- **Masquerading** is a form of SNAT where you let your firewall automatically detect the external interface address
- **SNAT** refers a situation where you explicitly specify what source address will be used when re-writing the outbound source address field.

The NAT table does not filter packets. Packet filtering is reserved for the **Filter Table**.

The NAT table utilizes the built-in PREROUTING, OUTPUT, and POSTROUTING rule chains.

## The Filter Table

The Filter table is the only table that is responsible for filtering packets; it should never alter them in the ways that the Mangle and NAT tables do, e.g., it should not alter the information in individual packets. The only work done by the Filter table consists of executing the targets ACCEPT, DROP, QUEUE, or RETURN.

**ACCEPT** means the packet continues traversing the chain.

**DROP** quietly drops the packet, without notifying the sender.

**QUEUE** passes the packet to userspace, where it may be picked up by the Mangle table, or may be passed along to other userspace utilities or modules.

**RETURN** sends the packet back to the rule following the last rule it passed in the **previous** rule chain; that is, when a rule is forwarded from one rule chain to another, the RETURN target will send a packet back to the next in the rule chain from which it was forwarded.

In addition, there one target extension may also be used with the Filter table:

**Reject** will drop the packet, but send an ICMP notification to the sending machine that the packet has been dropped.

The Filter table uses the built-in INPUT, OUTPUT, and FORWARD rule chains

## The Mangle Table

The Mangle table is primarily used to prioritize certain connections for quality of service optimizations; it is used for general packet header modification, such as setting the Time-to-Live (TTL) or Type-of-Service (TOS) fields, or to set an internal mark (called **nmark**, and set with the **MARK target**) to identify the packet for later processing.

# Understanding Rule Chains

## The Five Built-In Rule Chains

The tables handle five built-in chains:

1. All inbound packets hit the **PREROUTING** chain, with no exceptions. Any changes performed on the packets here are done before the routing decision and filtering is done. When connections are bound for machines located on the local subnet this chain will alter the destination IP address address for **destination address**

- translation** (DNAT). By the time a packet reaches the PREROUTING chain, all checks on the IP headers have been completed, but the packet has not yet been routed.
- The **INPUT** chain receives all **inbound** packets which are addressed to the local intranet served by this firewall. All packets which are addressed to the local intranet will be filtered here, before they continue onwards.
  - The **FORWARD** chain receives and filters all packets which are addressed to computers which are not located on the local intranet located behind the firewall, i.e., it redirects packets which are intended to be forwarded to other parts of the network which are not located on the subnet administered by the firewall, or which have arrived from sections of the network (not located behind the administered subnet) and are destined for the open Internet.
  - The **OUTPUT** chain receives all **outbound** packets which are addressed to computers outside the local intranet. All packets which are addressed to the local intranet served by the firewall will be filtered here, before they continue onwards, onto the Internet.
  - The **POSTROUTING** chain is the very last chain that is applied; all outbound packets which are leaving the local machine (or subnet) will pass through this chain. Packets which are processed by the POSTROUTING chain have already been routed, but have not been sent over the Ethernet. This is where Netfilter performs **source address translation** (SNAT), altering the source address from the IP address that is used on the local intranet to the one which identifies the firewall on the open Internet.

## User-Defined Chains

User-defined chains are used to create customized filters for a wide variety of needs; however, there are some commonly used chains which most administrators call when building a firewall. One example follows:

```
Moxa:~moxa@MOXA:~# iptables -N TCP && iptables -N UDP
```

This creates a user-defined chain called TCP and another called UDP, which you may use to manage protocols later on. To see how to implement these chains in the INPUT chain, see below, **Rule Examples: Applying User-Defined Chains**.



### ATTENTION

To find out what rules are currently written into each table and chain, use the commands described below, in the section **Viewing and Manipulating Rulesets**.

## Understanding Basic Traffic Flows

Users should recognize that these five chains may be used to build three fundamental traffic flows. Additionally, certain chains are only associated with certain tables. For more information on which tables use which chains, see the next section,

- A) **Forwarded** packets will traverse this set of chains in the following order:

```
PREROUTING → FORWARD → POSTROUTING
(in the NAT table) (in the Filter table) (in the NAT table)
```

- B) **Inbound** traffic that is destined for the local subnet will traverse this set of chains:

```
PREROUTING → INPUT → INPUT
(in the NAT table) (in the Mangle table) (in the Filter table)
```

- C) **Outbound** traffic that is leaving the firewall will traverse this set of chains:

```
OUTPUT → OUTPUT → POSTROUTING
(in the NAT table) (in the Mangle table) (in the Filter table)
```

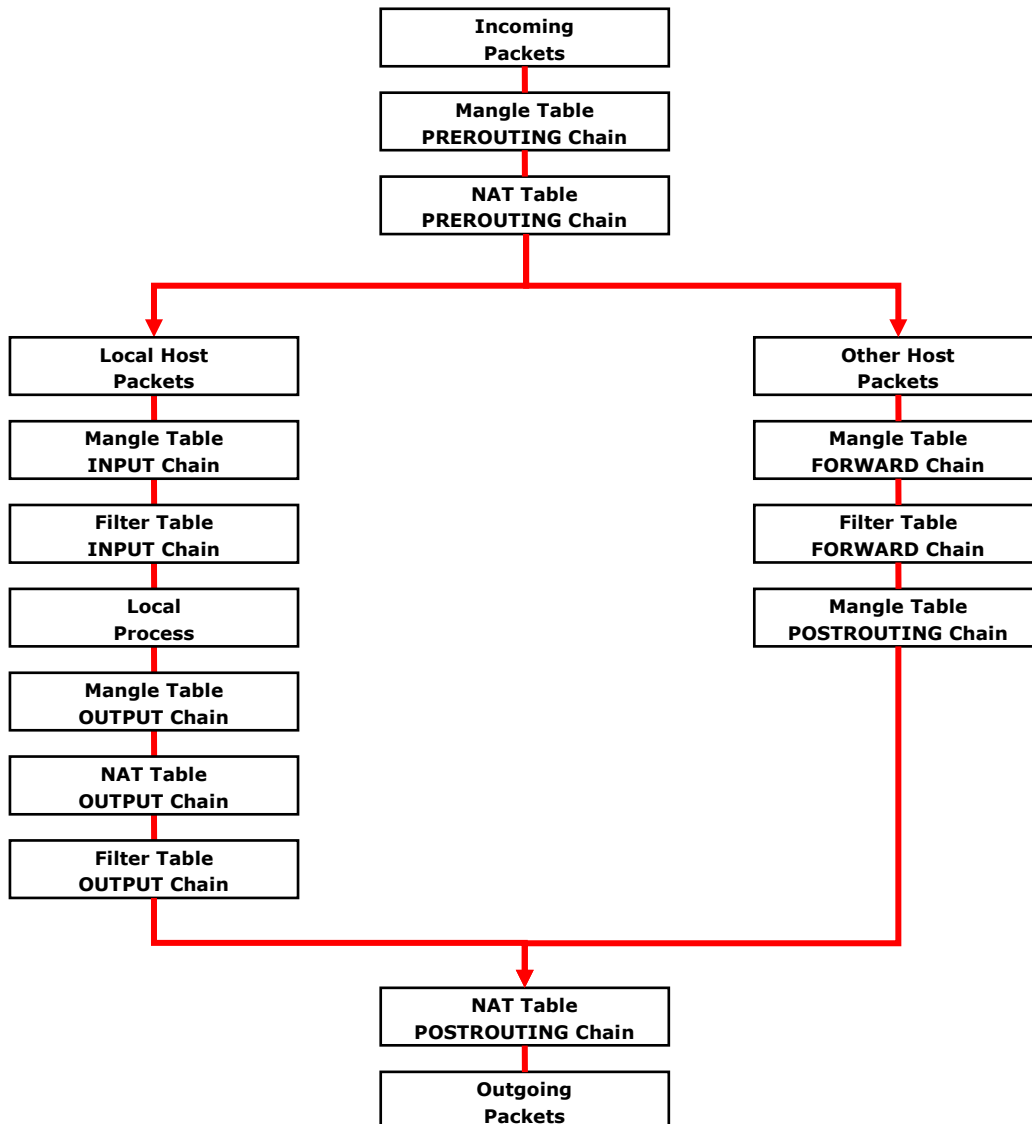


### ATTENTION

Building complex firewalls using the Netfilter rules and interface can become overwhelming, even for experienced administrators. If you require advanced firewall capabilities, Moxa recommends using a Netfilter configuration interface. One of the easiest to learn and most powerful is the Shorewall Firewall. Shorewall is available as a standard Debian package, and may be downloaded using apt-get. Shorewall documentation is available at the Shorewall website, found at <http://www.shorewall.net>.

## Netfilter Hierarchy for Incoming Packets

This figure shows how packets traverse the table hierarchy. Outbound packets originating on the local network start at the box labeled **Local Process**. Inbound packets start at the top box labeled **Incoming Packets**.



### ATTENTION

Be careful when setting up iptables rules. Incorrectly configured rules can very easily break connectivity with a remote host. For simple setups requiring minimal configuration (five rules or less), Moxa recommends directly configuring iptables using the console and a standard editor. For more complicated setups, users may use Arno's iptables firewall script, or for very large, extremely complicated setups Moxa recommends the Shoreline Firewall. The following links will take you to further information about iptables setups and the various software packages mentioned above.

The netfilter/iptables Project Homepage: <http://www.netfilter.org/index.html>

The Official netfilter/iptables packet-filtering HOWTO:

<http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.htm>

Arno's iptables Firewall (click on IPTABLES FIREWALL tab at the top navigation ribbon):

<http://rocky.eld.leidenuniv.nl/joomla/>

The Shorewall Firewall Homepage (lots of information about netfilter/iptables, as well):

[http://www.shorewall.net/Documentation\\_Index.html](http://www.shorewall.net/Documentation_Index.html)

Public iptables/netfilter Forum: <http://www.linuxguruz.com/iptables/>

## Connection Tracking

A connection tracking system does not filter packets. The Netfilter connection tracking system monitors kernel memory structures to keep track of the state of each connection; this means that it logs the protocol types, port number pairs, and source and destination IP addresses, and associates that with various connection states and timeout values. By being able to track connection states, it is possible to build much more powerful and secure filtering rules.

There are four states that may be defined for a connection:

- **NEW**  
This is the state when a connection is just initiating: the firewall has only seen traffic in one direction (either inbound or outbound) and if the packet is a valid one for initiating a connection (i.e., a SYN packet for a TCP request).
- **ESTABLISHED**  
This is used to describe a connection that has been successfully negotiated, and packets are being exchanged in both directions.
- **RELATED**  
At the application layer there are some protocols—like FTP passive mode, for instance—which are difficult to track. FTP passive mode uses a wide range of ports, from 1024 to 65535, rather than just one; tracking in these connections is much more difficult than simply tracking a connection across a single port (typically port 20, in FTP). The connection tracking system defines an expectation, which is a connection that is expected to happen in a set period of time, but that has a limited lifetime. Using helpers and expectations, the Netfilter connection tracking system is able to track connections according to patterns by defining **master** connections, and **related** connections.
- **INVALID**  
This is used to identify packets that do not follow the expected behavior of a connection. System administrators can set filters to drop them.

## Building the Firewall

The most secure systems begin by dropping all traffic and then adding in exceptions to allow network traffic in. This should be your philosophy, when building a firewall.

### Policies: Setting Default Firewall Behavior

Netfilter **policies** set the default behavior for its built-in tables, and policies may only be set for Netfilter's built-in tables. This means that policies set the default behavior for all packets handled by the firewall: if a packet arrives which no rule can process, Netfilter will default to the root policy set for that connection. Policies may be set for every table and chain, which means that default policies may be independently set for inbound, outbound, and forwarded packets.

The default policy for most firewalls should be an across-the-board **drop** all connections; after setting the policies to drop all connections, administrators may then add exceptions to allow connections through on a case-by-case basis. This section will only show you how to set the policies; to see how to write rules, look at the section below, [Writing Rulechains](#).



#### **WARNING**

Firewall rules are only valid for the time the computer is on. If the system is rebooted, the rules will be automatically flushed. To save a ruleset so that it loads on the next reboot, use the following command:

```
moxa@moxa:~# /sbin/service iptables save
```

## Setting Policies

```
Moxa:~moxa@MOXA:~# iptables [-t tables] [-P, --policy chain target] [Policy: ACCEPT, DROP, ETC]
```

### Command Arguments:

**-P, --policy:** This sets a default policy the firewall will enforce on a particular chain for a particular table. Only built-in chains (i.e.: not user-defined) can have policies. Possible targets for policy enforcement are INPUT, OUTPUT, FORWARD, PREROUTING, OUTPUT, and POSTROUTING. Possible policies that may be enforced on these chains are ACCEPT, DROP, QUEUE, and RETURN (see below for explanation).

**INPUT:** Targets packets coming into the V2101-LX over the **filter**, **mangle**, or **security** tables.

**OUTPUT:** Targets locally-generated packets leaving the V2101-LX. All tables have an output chain.

**FORWARD:** Targets packets routed through the machine, on the **filter**, **mangle**, or **security** tables.

**PREROUTING:** Targets packets for alteration before they have traversed the firewall; used on the **NAT**, **mangle**, and **raw** tables.

**POSTROUTING:** Targets packets as they are about to be sent out over the **NAT** and **mangle** tables.

### Policy Arguments:

**ACCEPT:** By default, all packets are let through the chain.

**DROP:** Packets are dropped, with no notification or response sent back to the originating computer.

**QUEUE:** Passes the packet to userspace; see **NFQUEUE** in Netfilter/iptables documentation for more information about how these targets are used.

**RETURN:** Stop traversing this chain and resume at the next rule in the previous (calling) chain.

**REJECT:** Equivalent to DROP, but it returns a message to the packet's origin.

**LOG:** Turns on kernel logging for matching packets, printing information on all matching packets on the kernel log where it may be read using *dmesg* or *syslogd*.

### Netfilter Policy Examples:

```
Moxa:~moxa@MOXA:~# iptables -P INPUT DROP
```

This changes the default policy so that **all incoming packets** on **all chains** are **dropped**, with no notification. This is Moxa's recommended setting for the input interface.

```
Moxa:~moxa@MOXA:~# iptables -P OUTPUT ACCEPT
```

This rule accepts **all outgoing packets** that originate on the local network, and is acceptable for a strictly secure internal network. If you change this policy to DROP it will considerably increase the complexity of the firewall. However, you may wish to consider this for computers that will be serving data to untrusted clients (such as customers, on a local network). For instance, to guarantee security on a train computer that will be serving wireless connections (that originate outside the train) to local passengers, the default rule should be **DROP**, with only specific, secure protocols and services allowed through on a rule-by-rule basis.



### ATTENTION

A useful tool for building more complex firewalls is the Shoreline Firewall, or Shorewall. The Shorewall homepage contains full documentation and software downloads, and may be found at the following URL:

<http://www.shorewall.net>

```
Moxa:~moxa@MOXA:~# iptables -P FORWARD DROP
```

This sets the FORWARD chain in the filter table to **DROP** all packets. **This is the recommended policy for all firewalls**, and may be safely used on devices occupying a terminal segment in the network topology, this is the appropriate rule.

```
Moxa:~moxa@MOXA:~# iptables -t nat -P PREROUTING ACCEPT
```

The nat tables are for address translation, not for filtering. The **PREROUTING** chain for the **NAT** should be set to **ACCEPT**, otherwise connection initialization packets will not be able to get through the firewall.

```
Moxa:~moxa@MOXA:~# iptables -t nat -P OUTPUT ACCEPT
```

The nat tables are for address translation, not for filtering. The **OUTPUT** chain for the NAT should be set to **ACCEPT**, otherwise connection initialization packets will not be able to get through the firewall.

```
Moxa:~moxa@MOXA:~# iptables -t nat -P POSTROUTING ACCEPT
```

The nat tables are for address translation, not for filtering. The **POSTROUTING** chain for the NAT should be set to **ACCEPT**, otherwise connection initialization packets will not be able to get through the firewall.

## Viewing and Manipulating Rulesets

Beginning with this section you will be provided some examples of rules commonly used to manipulate, view, and configure simple firewalls for industrial environments. For simple setups, typically only three or four rules are needed to give a device strong protection against unauthorized network intrusions.

### List current rule chains for a target table, or for all tables

The full command for **listing** rule chains is as follows:

```
MOXA:~moxa@MOXA:~# iptables [-t table, or multiple, tables,...] [-L chain] [-n]
```

#### Command Arguments:

- t: Table to manipulate (default: 'filter'); available args are **filter**, **nat**, **mangle**, **raw**, and **security**
- L: Indicates a chain to be listed. If no chain is selected, all chains are listed.
- n: Returns the numeric output of addresses and ports: e.g. TCP and UDP ports are printed as numbers, rather than names. This also saves execution time by preventing iptables from looking up DNS requests.



#### WARNING

Simple commands listing iptable NAT or filter rules will autoload selected kernel modules, including the connection tracking (conntrack) and filter (iptable\_filter) modules. On high-capacity production servers, these modules easily overload and bring the networking system down. Whenever a list command is issued, check the message buffer (**dmesg**) to see if drivers have been auto-loaded, and what they are. For more information, see <http://backstage.soundcloud.com/2012/08/shoot-yourself-in-the-foot-with-iptables-and-kmod-auto-loading/>.

### Flush a current rule chain, or delete a user-specified chain

The full command to **flush** rule chains is as follows:

```
MOXA:~moxa@MOXA:~# iptables [-t table, or tables] [-FXZ]
```

#### Command Arguments:

- t: Table to manipulate; choices are **filter**, **nat**, **mangle**, **raw**, and **security**. Defaults to **filter**.
- F: Flush the selected chain (if no chains are specified, this flushes all the chains in the table)
- X: Delete the specified user-defined chain (chain must be empty and all references to the chain must be deleted first); if no argument is given, all non-built-in chains will be deleted



#### WARNING

The command `moxa@MOXA:~moxa@MOXA:~# iptables -F` will flush all iptables rulechains from the kernel, permanently deleting the firewall and fully exposing the computer to the open Internet. IPTables rules will not automatically reload following a restart; you must configure your machine to reload them, either using a shell script or some other firewall package.

You should save your configured firewall in a file so you can conveniently re-load it. The following command will save all of the current iptables rules to the file `/etc/iptables.save` (you may name the file whatever you wish):

```
moxa@MOXA:~moxa@MOXA:~# iptables-save > /etc/iptables.save
```

You may load the rules contained in this file using the **iptables-restore** command, as follows:

```
moxa@MOXA:~moxa@MOXA:~# iptables-restore < /etc/iptables.save
```

## Zero-out the packet and byte counters for a rule chain

Zeroing the counters is sometimes useful when monitoring firewall activity for analysis. When used in combination with the list argument, the zero argument will give a precise measurement of the number of packets that have been processed since the last measurement, for all chains, a given chain, or even a given rule within a chain. The full command to **flush** rule chains is as follows:

```
moxa@MOXA:~moxa@MOXA:~# iptables -L -Z -n [chain [rulenum]]
```

### Command Arguments:

-Z: Set the packet and byte counters to zero in all chains, for only a given chain, or only a rule in a chain

## Delete a User-Generated Chain

This command deletes a specified user-defined chain.

```
moxa@MOXA:~moxa@MOXA:~# iptables -X [chain]
```

There must be no references to the chain in other chains or tables, and the chain must be empty, i.e. not contain any rules. You must delete or replace any remaining referring rules before the chain can be deleted. *If no argument is given, this will attempt to delete every user defined chain in the table.*

## Writing Rulechains

In this section we show you how to write rules for a simple industrial network firewall. More complicated firewalls—such as those serving public networks, or untrusted customers—are beyond the scope of this manual. For advanced firewall needs, Moxa recommends the use of the Shoreline Firewall (Shorewall) mentioned above.

```
MOXA:~moxa@MOXA:~# iptables [-t table] [-AI] [INPUT, OUTPUT, FORWARD] [-io interface] /
    [-p tcp, udp, icmp, all] [-s IP/network] [--sport ports] [-d IP/network] /
    [--dport ports] -j [ACCEPT. DROP]
```

-A: **Append** one or more rules to the end of the selected chain

-I: **Insert** one or more rules in the selected chain as the given rule number

-i: Identifies an **interface** which will **received** a packet

-o: Identifies an **interface** over which a packet will be **sent**

-p: Identifies the **protocol** to be filtered

-s: Identifies a **source address** (network name, host name, network IP address, or plain IP address)

--sport: Identifies the **source port**, or the port where the packet originated

-d: Identifies the **destination address** (network name, host name, NAT or IP address)

--dport: Identifies the **destination port**, or the port where the packet will terminate

-j: Jump target. Specifies the target of the rules; i.e., how to handle matched packets.

For example, ACCEPT the packet, DROP the packet, or LOG the packet.



### WARNING

For all firewalls using a strict DROP policy on incoming packets, be sure to include a rule that accepts packets on the loopback interface:

```
moxa@MOXA:~moxa@MOXA:~# iptables -A INPUT -i lo -j ACCEPT
```

### Examples:

**REQUIRED RULE** for all firewalls:

Accept all packets from the loopback interface:

```
moxa@MOXA:~# iptables -A INPUT -i lo -j ACCEPT
```

**RECOMMENDED RULE** from the **sample firewall** provided in **Appendix C: Sample Scripts**:

Allow all traffic from that belongs to established connections, or new, related traffic:

```
moxa@MOXA:~# iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

**RECOMMENDED RULE** from the **sample firewall** provided in **Appendix C: Sample Scripts:**

Drops all traffic with an invalid state, (e.g. **Port Unreachable**) when nothing was sent to the host, invalid headers or checksums, and out-of-sequence packets:

```
moxa@MOXA:~# iptables -A INPUT -m conntrack --ctstate INVALID -j DROP
```

**Basic Filter Rules** show examples of how you can open commonly opened ports:

Web server / HTTP:

```
moxa@MOXA:~# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Secure-sockets web server / HTTPS:

```
moxa@MOXA:~# iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

Remote SSH Connections (**REQUIRED RULE**):

```
moxa@MOXA:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Incoming UDP Streams:

```
moxa@MOXA:~# iptables -A INPUT -p udp --dport 53 -j ACCEPT
```

**ATTENTION**

ICMPv6 Neighbor Discovery packets will always be classified INVALID (if you don't know what this means, you can probably ignore it). You may accept them with this rule:

```
moxa@MOXA:~# iptables -A INPUT -p 41 -j ACCEPT
```

Example 1: ACCEPT all TCP packets from 192.168.0.1.

```
moxa@MOXA:~# iptables -A INPUT -i eth0 -p tcp -s 192.168.0.1 -j ACCEPT
```

Example 2: ACCEPT all TCP packets from Class C network 192.168.1.0/24.

```
moxa@MOXA:~# iptables -A INPUT -i eth0 -p tcp -s 192.168.1.0/24 -j ACCEPT
```

Example 3: DROP all TCP packets from 192.168.1.25 (this rule is only necessary on firewalls where you have set the INPUT policy to ACCEPT; **this is not recommended**).

```
moxa@MOXA:~# iptables -A INPUT -i eth0 -p tcp -s 192.168.1.25 -j DROP
```

Example 4: ACCEPT all TCP packets addressed for port 21.

```
moxa@MOXA:~# iptables -A INPUT -i eth0 -p tcp --dport 21 -j ACCEPT
```

Example 5: ACCEPT all packets from 192.168.0.24 destined for ports 137, 138, and 139 on this computer.

```
moxa@MOXA:~# iptables -A INPUT -i eth0 -p tcp -s 192.168.0.24 --dport 137:139 -j ACCEPT
```

Example 7: LOG all TCP packets destined for port 25.

```
moxa@MOXA:~# iptables -A INPUT -i eth0 -p tcp --dport 25 -j LOG
```

**ATTENTION**

To use the rule in Examples 8 and 9, below, remember to first load the module **ipt\_mac**:

```
moxa@MOXA:~# modprobe ipt_mac.
```

To make a module load across reboots, you may add it to the `/etc/modprobe.conf` file using this command:

```
moxa@MOXA:~# echo "ipt_mac" >> /etc/modprobe.conf
```

Don't forget to backup your `modprobe.conf` file before altering it, and take care to use the double pointer (`>>`)—which is **append**—rather the single pointer (`>`) which is **overwrite**.

Example 8: Drop all packets from MAC address 01:02:03:04:05:06.

```
moxa@MOXA:~# iptables -A INPUT -i eth0 -p all -m mac --mac-source 01:02:03:04:05:06 -j DROP
```

Example 9: Accept all packets from MAC address 02:03:04:05:06:07.

```
moxa@MOXA:~# iptables -A INPUT -i eth0 -p all -m mac --mac-source 02:03:04:05:06:07 -j ACCEPT
```

**Rule Examples: Applying User-Defined Chains**

Some network administrators may find it useful to define their own rule chains. Here, we show how to implement them in the INPUT chain, and use the chains defined above, in the section **User-Defined Chains**.



```
moxa@MOXA:~# iptables -A INPUT -p udp -m conntrack --ctstate NEW -j UDP
moxa@MOXA:~# iptables -A INPUT -p tcp --syn -m conntrack --ctstate NEW -j TCP
```

The TCP and UDP chains are now attached to the INPUT chain; by adding in the above connection rule, once a connection is accepted by either chain, it will be handled by the RELATED/ESTABLISHED rule. You may now add rules to these chains as if you were adding rules to the INPUT chain. Using some of the INPUT rules defined above as examples:

```
moxa@MOXA:~# iptables -A TCP -p tcp --dport 80 -j ACCEPT
moxa@MOXA:~# iptables -A TCP -p tcp --dport 443 -j ACCEPT
moxa@MOXA:~# iptables -A TCP -p tcp --dport 22 -j ACCEPT
moxa@MOXA:~# iptables -A UDP -p udp --dport 53 -j ACCEPT
```



### ATTENTION

A sample firewall is provided in **Appendix A: A Sample Firewall**.

## Setting Up NAT (Network Address Translation)

The NAT (Network Address Translation) protocol translates IP addresses used on a local network into IP addresses used on a connecting network. One network is designated the inside network and the other is the outside network. Typically, the V2101-LX connects several devices on a network and maps local inside network addresses to one or more global outside IP addresses, and translates the global IP address used on by packets coming in from the WAN back into local IP addresses.

### IP Tables NAT Policies

IP tables policies for the NAT table should all be ACCEPT (see the section above, **Netfilter Policy Examples**, for more information):

```
moxa@MOXA:~# iptables -t nat -P PREROUTING ACCEPT
moxa@MOXA:~# iptables -t nat -P POSTROUTING ACCEPT
moxa@MOXA:~# iptables -t nat -P OUTPUT ACCEPT
```

### Source NAT (SNAT) and Destination NAT (DNAT)

**Source NAT (SNAT)** is when the source address is altered on the first packet of an outbound connection. That is, it changes the originating address (which is usually a LAN address that looks like 192.168.xxx.xxx) for outbound packets so that they show the IP address with which the connection to the open internet is associated.

**Destination NAT (DNAT)** is when the destination address is altered on the first packet of an outbound connection. That is, it changes the originating address (which is usually a LAN address that looks like 192.168.xxx.xxx) for outbound packets so that they show the IP address with which the connection to the open internet is associated.



### ATTENTION

Click on the following link for more information about NAT:

<http://www.netfilter.org/documentation/HOWTO/NAT-HOWTO.html>

### Enabling NAT Masquerading

NAT masquerading allows you to create a subnet of devices mapped to a single IP address. When used with port forwarding and static IP addressing, it can allow you to expand a single public IP address to a very large LAN.

To enable NAT in your device, first load the NAT module:

```
moxa@MOXA:~# modprobe ipt_MASQUERADE
```



### ATTENTION

To make a module load across reboots, you may add it to the `/etc/modprobe.conf` file using this command:

```
moxa@MOXA:~# echo "ipt_MASQUERADE" >> /etc/modprobe.conf
```

Don't forget to backup your `modprobe.conf` file before altering it, and take care to use the double pointer (`>>`)—which is **append**—rather the single pointer (`>`) which is **overwrite**.

In the **NAT** table (`-t nat`), **Append** a rule (`-A`) after routing (**POSTROUTING**) all packets going out **ppp0** (`-o ppp0`) which says to **MASQUERADE** the connection (`-j MASQUERADE`).

```
moxa@MOXA:~# iptables -t nat -A POSTROUTING -o eth0 -s 555.666.777.888/24 -j MASQUERADE
```

Then turn on IP forwarding:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Using these rules and DHCP, it will now be possible to allow local devices to communicate with devices outside the subnet; however, communications will only be able to be initiated from the local network. To allow full address translation both ways, you will need to set up static IP addresses for your devices, and port forwarding rules. These are beyond the scope of this manual, but you can find many rule examples in **Appendix A: A Sample Firewall**.

## Saving the Firewall

*You must save your firewall configuration so that it will reload on the next reboot;* otherwise, the rules will be flushed and the firewall permanently deleted. After configuring iptables, the following command will save the ruleset to `/etc/sysconfig/iptables`:

```
moxa@MOXA:~# iptables-save > /etc/Active-Firewall-Rules_YYYY-MM-DD.filter
```

The most convenient way of guaranteeing that your iptables will automatically reload following a restart is through the **iptables-persistent** package. This is a simple script that will record the current iptables configuration to a set of rules at `/etc/iptables/rules.v4`. However, administrators will need to re-run the script every time they update or change the netfilter rules. You may do this by either re-installing the package (using `apt` or `aptitude`), or by using the `dpkg-reconfigure` command as below:

```
moxa@MOXA:~# dpkg-reconfigure iptables-persistent
```

Another way of setting the firewall to automatically load is by adding two lines (shown below) to your `/etc/rc.local` file, or to the `/etc/network/interfaces/if-up.d/upstart` file:

```
# Load iptables rules from this file
iptables-restore < /etc/iptables.conf
```

## Setting up a Networked File System: NFS

The Network File System (NFS) is used by client computers to mount a remote disk partition as if it were part of their local hardware. NFS is a distributed file system that allows fast, seamless sharing of files across a network. NFS allows users to develop applications for the V2101-LX without worrying about the amount of disk space that will be available. The V2101-LX only supports NFS client protocol.

The following procedures illustrate how to mount a remote NFS Server. **192.168.3.5**, in step 3, is the IP address of the NFS server.

7. Scan the NFS Server's shared directory:

```
moxa@MOXA:~# showmount -e HOST

showmount:          Shows the mount information of an NFS Server
-e:                 Shows the NFS Server's export list.
HOST:               IP address or DNS address
```

8. Establish a mount point on the NFS Client site:

```
moxa@MOXA:~# mkdir -p /home/nfs/public
```

9. Mount the remote directory to a local directory:

```
moxa@MOXA:~# mount -t nfs -o nolock 192.168.3.5:/home/public /home/nfs/public
(192.168.3.5 is the example IP address of the NFS server.)
```



### ATTENTION

To set up a mount process to mount at boot-time, copy the mount command into the `/etc/fstab` file. For more information on NFS and its configuration options, you may refer to the NFS homepage, at: <http://nfs.sourceforge.net/> (Dec. 2013).

## Setting Up a VPN

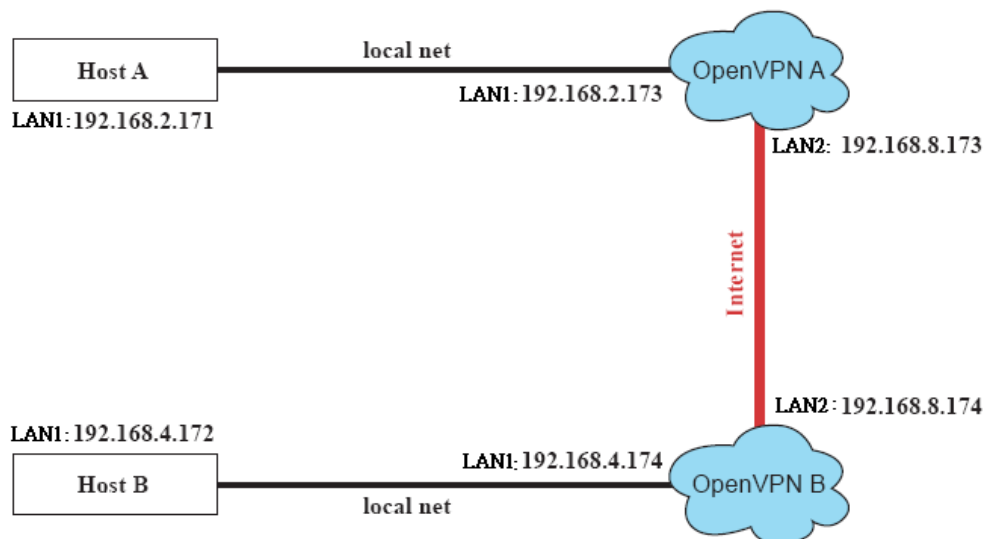
This platform uses the OpenVPN package to provide VPN capability. OpenVPN provides two basic types of tunnels for users to implement VPNS: **Routed IP Tunnels** and **Bridged Ethernet Tunnels**.

An Ethernet bridge is used to connect different Ethernet networks together. The Ethernets are bundled into one bigger, logical network that can communicate securely across the open Internet. Each Ethernet corresponds to one physical interface (or port) that is connected to the bridge.

On each OpenVPN machine, you should carry out configurations in the `/etc/openvpn` directory, where script files and key files reside. Once established, all operations will be performed in that directory.

### Ethernet Bridges Linking Independent Subnets Over the Internet

This setup will link at two independent subnets over the Internet. It will use at least four machines, as shown in the following diagram. **OpenVPN** designates a dedicated VPN server (perhaps also a firewall), while **Host** designates a client computer located behind the VPN server.



**Host A** represents the machine that belongs to the subnet served by the VPN server, **OpenVPN A**, and **Host B** represents a machine that belongs to the subnet served by the VPN server, **OpenVPN B**. The two remote subnets are configured for **distinct ranges of IP addresses** on **separate subnets**. When this configuration is moved to a public network, the external interfaces of the OpenVPN machines must be configured for static IPs, or connected to another device (such as a firewall or DSL box) that uses a static address. To set up a bridged Ethernet tunnel following this basic architecture, follow the instructions below:

1. Generate a preset shared key by typing the following command:  

```
moxa@MOXA:~# openvpn --genkey --secret secrouter.key
```
2. Copy the keyfile that you have just generated to the OpenVPN machines:  

```
moxa@MOXA:~# scp /etc/openvpn/secrouter.key XXX.XXX.X.XXX:/etc/openvpn
```

**ATTENTION**

Select cipher and authentication algorithms by specifying cipher and auth. To see which algorithms and ciphers are available, type:

```
moxa@MOXA:~# openvpn --show-ciphers
moxa@MOXA:~# openvpn --show-auths
```

For testing purposes, a preshared key is provided at `/etc/openvpn/secrouter.key`. This is adequate for testing, but users must create a new key when going live or their network will be insecure..

**Configuring OpenVPN A: VPN Server**

1. Modify the remote address in the configuration file `/etc/openvpn/tap0-br.conf` by adding the IP address for the remote server (in this case, OpenVPN B).

```
# point to the peer
remote 192.168.8.174
    dev tap0
    port 1194
    secret /etc/openvpn/secrouter.key
    cipher DES-EDE3-CBC
    auth MD5
    tun-mtu 1500
    tun-mtu-extra 64
ping 40
up /etc/openvpn/tap0-br.sh
#comp-lzo
```

2. Next, modify the routing table in `/etc/openvpn/tap0-br.sh` script so that it maps the internal subnet VPN server A will be serving.

```
#-----Start-----
#!/bin/sh
# value after "-net" is the subnet behind the remote peer
route add -net 192.168.4.0 netmask 255.255.255.0 dev br0
#-----end-----
```

3. And then configure the bridge interface in `/etc/openvpn/bridge`.

```
#!/bin/bash
# Create global variables
# Define Bridge Interface
br="br0"
# Define list of TAP interfaces to be bridged,
# for example tap="tap0 tap1 tap2".
tap="tap0"
# Define physical ethernet interface to be bridged
# with TAP interface(s) above.
eth="eth1"
eth_ip="192.168.8.173"
eth_netmask="255.255.255.0"
eth_broadcast="192.168.8.255"
#gw="192.168.8.174"
...
```

4. Start the VPN link by calling the bridge script:

```
moxa@MOXA:~# /etc/openvpn/bridge restart
```

## Configuring OpenVPN B: VPN Subnet Client

5. Modify the **remote address** entry in the VPN configuration file, `/etc/openvpn/tap0-br.conf`.

```
# point to the peer
remote 192.168.8.173
dev tap0
secret /etc/openvpn/secrouter.key
  cipher DES-EDE3-CBC
  auth MD5
  tun-mtu 1500
  tun-mtu-extra 64
ping 40
up /etc/openvpn/tap0-br.sh
#comp-lzo
```

6. Next modify the routing table in the `/etc/openvpn/tap0-br.sh` script file.

```
#-----Start-----
#!/bin/sh
# value after "-net" is the subnet behind the remote peer
route add -net 192.168.2.0 netmask 255.255.255.0 dev br0
#----- end -----
```

7. And then configure the bridge interface script in `/etc/openvpn/bridge`.

```
#!/bin/bash
# Create global variables
# Define Bridge Interface
br="br0"
# Define list of TAP interfaces to be bridged,
# for example tap="tap0 tap1 tap2".
tap="tap0"
# Define physical ethernet interface to be bridged
# with TAP interface(s) above.
eth="eth1"
eth_ip="192.168.8.174"
eth_netmask="255.255.255.0"
eth_broadcast="192.168.8.255"
#gw="192.168.8.173"
...
```

8. Start the bridge script file to configure the bridge interface.

```
moxa@MOXA:~# /etc/openvpn/bridge restart
```

9. Start the OpenVPN peers that are on machine OpenVPN A and OpenVPN B with the following command:

```
moxa@MOXA:~# openvpn --config /etc/openvpn/tap0-br.conf&
```

If you see a line that looks like **Peer Connection Initiated with 192.168.8.173:5000** on each machine, then the connection the Ethernet bridge has been successfully established over UDP port 5000.

10. Check the routing table on each VPN server by typing the command below:

```
moxa@MOXA:~# route
```

Destination	Gateway	Genmsk	Flags	Metric	Ref	Use	Iface
192.168.5.0	0.0.0.0	255.255.255.0	U	0	0	0	eth2
192.168.4.0	0.0.0.0	255.255.255.0	U	0	0	0	br0
192.168.3.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
192.168.30.0	0.0.0.0	255.255.255.0	U	0	0	0	eth3
192.168.8.0	0.0.0.0	255.255.255.0	U	0	0	0	br0

Interface **eth1** and device **tap0** both connect to the bridging interface, and the virtual device **tun** sits on top of **tap0**. This ensures that all traffic coming to this bridge from internal networks connected to interface **eth1** write to the TAP/TUN device that the OpenVPN program monitors. Once the OpenVPN program detects traffic on the virtual device, it sends the traffic to its peer.

11. To create an indirect connection to Host B from Host A, you need to add the following routing item:
- ```
moxa@MOXA:~# route add -net 192.168.4.0 netmask 255.255.255.0 dev eth0
```

To create an indirect connection to Host A from Host B, you need to add the following routing item:

```
moxa@MOXA:~# route add -net 192.168.2.0 netmask 255.255.255.0 dev eth0
```

Now ping Host B from Host A by typing:

```
moxa@MOXA:~# ping 192.168.4.174
```

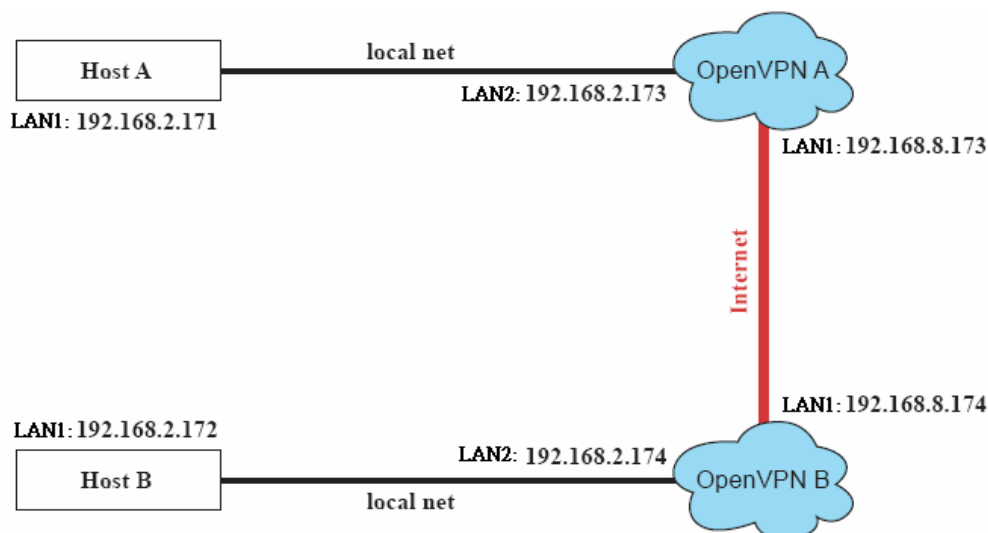
A successful ping indicates that you have created a VPN system that only allows authorized users from one internal network to access users at the remote site. For this system, all data is transmitted by UDP packets on port 5000 between OpenVPN peers.

12. To shut down the VPN servers, use the **killall** command:

```
moxa@MOXA:~# killall -TERM openvpn
```

## Ethernet Bridging for Private Networks on the Same Subnet

Like the last example, this setup will link two subnets across the open Ethernet; however, these two subnets will share addressing as if they were located on the same local subnet.



All of the clients on the two remote subnets are configured for a range of IP addresses that spans **the same subnet**. When this configuration is moved to a public network, the external interfaces of the OpenVPN machines must be configured for static IPs or connected to another device (such as a firewall or DSL box) that uses a static address.

The configuration procedure for this setup is almost the same as for the previous example. The only difference is that you will need to comment out the parameter **up** in the `/etc/openvpn/tap0-br.conf` on each of the gateways, OpenVPN A and OpenVPN B.

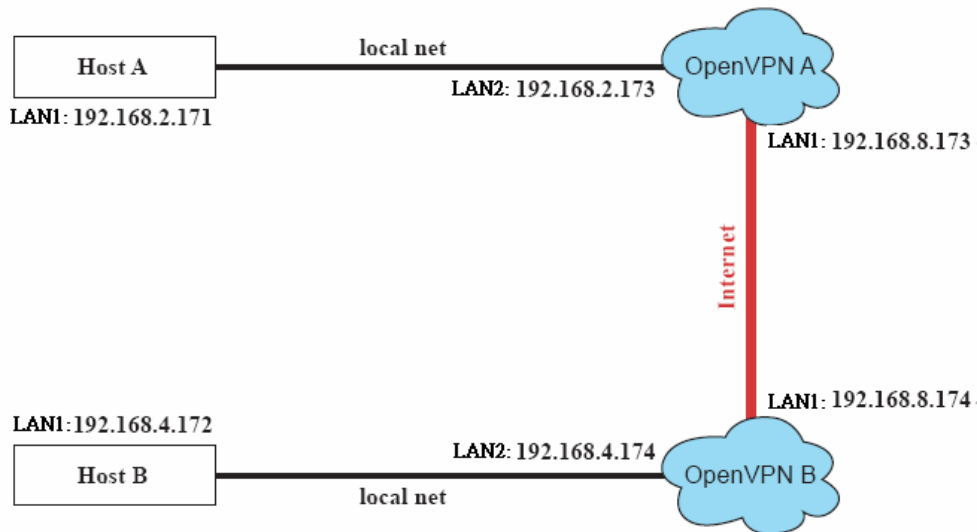
```
# point to the peer
remote 192.168.8.174
dev tap0
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
```

```
ping 40
#up /etc/openvpn/tap0-br.sh
#comp-lzo
```

## Routed IP Tunnels

Routed IP tunnels are used to route point-to-point IP traffic without broadcasts; the advantage of routed IP tunnels is that they are slightly more efficient than bridged ethernet tunnels and easier to configure.

3. **Host A** represents the machine that belongs to the subnet served by the VPN server, **OpenVPN A**, and **Host B** represents a machine that belongs to the subnet served by the VPN server, **OpenVPN B**. The two remote subnets are configured for **distinct ranges** of private IP addresses on **separate subnets**.



4. On VPN server A (**OpenVPN A**), modify the **remote address** entry in the configuration file `/etc/openvpn/tun.conf` by adding the address of OpenVPN B. Also, you must add an **ifconfig** entry which indicates the local (1<sup>st</sup>) and remote (2<sup>nd</sup>) VPN gateway addresses, separated by a space.

```
# point to the peer
remote 192.168.8.174
dev tun
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
ifconfig 192.168.2.173 192.168.4.174
up /etc/openvpn/tun.sh
```

5. Next, change OpenVPN A's `/etc/openvpn/tun.sh` so that the routing table matches the local subnet the VPN gateway is serving. Notice the **gw \$5** appended to the end of this line: the **\$5** is a variable argument that OpenVPN passes to the startup script. Its value is the second argument of **ifconfig** in the `/etc/openvpn/tun.conf` file.

```
#-----Start-----
#!/bin/sh
# value after "-net" is the subnet behind the remote peer
route add -net 192.168.2.0 netmask 255.255.255.0 gw $5
#-----end-----
```

6. On VPN server B (**OpenVPN B**), change the **remote address** in configuration file `/etc/openvpn/tun.conf` by adding the address of OpenVPN A. Also, you must add an **ifconfig** entry which indicates the local (1<sup>st</sup>) and remote (2<sup>nd</sup>) VPN gateway addresses, each separated by a space.

```
# point to the peer
remote 192.168.8.173
dev tun
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
ifconfig 192.168.4.174 192.168.2.173
up /etc/openvpn/tun.sh
```

7. Next, change OpenVPN B's routing table in the file `/etc/openvpn/tun.sh` so that it matches the local subnet the VPN gateway is serving. Notice the **gw \$5** appended to the end of this line: the **\$5** is a variable argument that OpenVPN passes to the script file. Its value is the second argument of **ifconfig** in the `/etc/openvpn/tun.conf` file.

```
#-----Start-----
#!/bin/sh
# value after "-net" is the subnet behind the remote peer
route add -net 192.168.2.0 netmask 255.255.255.0 gw $5
#-----end-----
```

8. Check the routing table after you run OpenVPN; it should show an established route running between your two VPN gateways. The command to see the routing table is:

```
moxa@Moxa:~# route.
```

| Destination   | Gateway       | Genmsk          | Flags | Metric | Ref | Use  |
|---------------|---------------|-----------------|-------|--------|-----|------|
| Iface         |               |                 |       |        |     |      |
| 192.168.4.174 | *             | 255.255.255.255 | UH    | 0      | 0   | tun0 |
| 192.168.4.0   | 192.168.4.174 | 255.255.255.0   | UG    | 0      | 0   | tun0 |
| 192.168.2.0   | *             | 255.255.255.0   | U     | 0      | 0   | eth1 |
| 192.168.8.0   | *             | 255.255.255.0   | U     | 0      | 0   | eth0 |

## SNMP (Simple Network Management Protocol)

The V2101-LX comes with the SNMP V1 (Simple Network Management Protocol) agent software pre-installed. It supports RFC1317 **RS-232 like group** and **RFC 1213 MIB-II**. The following shows example shows an SNMP agent responding to a query from the SNMP browser on the host site:

```
***** SNMP QUERY STARTED *****
[root@jaredRH90 root]# snmpwalk -v 1 -c public 192.168.30.128|more
SNMPv2-MIB::sysDescr.0 = STRING: Linux Moxa 2.6.18-5-686 #1 SMP Mon Dec 24 16:41
:07 UTC 2007 i686
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.8691.12.680
SNMPv2-MIB::sysUpTime.0 = Timeticks: (134544) 0:22:25.44
SNMPv2-MIB::sysContact.0 = STRING: "Moxa Inc."
SNMPv2-MIB::sysName.0 = STRING: Moxa
SNMPv2-MIB::sysLocation.0 = STRING: "Fl.8, No.6, Alley 6, Lane 235, Pao-Chiao Rd
. Shing Tien City, Taipei, Taiwan, R.O.C."
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORID.1 = OID: IF-MIB::ifMIB
```



```
SNMPv2-MIB::sysORID.2 = OID: SNMPv2-MIB::snmpMIB
SNMPv2-MIB::sysORID.3 = OID: TCP-MIB::tcpMIB
SNMPv2-MIB::sysORID.4 = OID: IP-MIB::ip
SNMPv2-MIB::sysORID.5 = OID: UDP-MIB::udpMIB
SNMPv2-MIB::sysORID.6 = OID: SNMP-VIEW-BASED-ACM-MIB::vacmBasicGroup
SNMPv2-MIB::sysORID.7 = OID: SNMP-FRAMEWORK-MIB::snmpFrameworkMIBCompliance
SNMPv2-MIB::sysORID.8 = OID: SNMP-MPD-MIB::snmpMPDCompliance
SNMPv2-MIB::sysORID.9 = OID: SNMP-USER-BASED-SM-MIB::usmMIBCompliance
SNMPv2-MIB::sysORDescr.1 = STRING: The MIB module to describe generic objects fo
r network interface sub-layers
SNMPv2-MIB::sysORDescr.2 = STRING: The MIB module for SNMPv2 entities
SNMPv2-MIB::sysORDescr.3 = STRING: The MIB module for managing TCP implementatio
...
SNMPv2-MIB::snmpOutBadValues.0 = Counter32: 0
SNMPv2-MIB::snmpOutGenErrs.0 = Counter32: 0
SNMPv2-MIB::snmpOutGetRequests.0 = Counter32: 0
SNMPv2-MIB::snmpOutGetNexts.0 = Counter32: 0
SNMPv2-MIB::snmpOutSetRequests.0 = Counter32: 0
SNMPv2-MIB::snmpOutGetResponses.0 = Counter32: 540
SNMPv2-MIB::snmpOutTraps.0 = Counter32: 0
SNMPv2-MIB::snmpEnableAuthenTraps.0 = INTEGER: disabled(2)
SNMPv2-MIB::snmpSilentDrops.0 = Counter32: 0
SNMPv2-MIB::snmpProxyDrops.0 = Counter32: 0
[root@jaredRH90 root]#
***** SNMP QUERY FINISHED *****
```

**ATTENTION**

Click on the following links for more information about RFC1317 RS-232 like group and RFC 1213 MIB-II.

<http://www.faqs.org/rfcs/rfc1317.html>

<http://www.ietf.org/rfc/rfc1213.txt>

# 4

## Programmer's Guide

---

The following topics are covered in this chapter:

- **Device API**
- **RTC (Real Time Clock)**
- **UART**
- **Digital I/O**

## Device API

The V2101 supports control devices with the **ioctl** system API. The **ioctl()** function manipulates the underlying device parameters of special files. Many operating characteristics of **character special files** (i.e., **character devices**, or **terminals**) may be controlled with **ioctl()** requests. The argument **d** must be an open file descriptor. The second argument is a **device-dependent request code**. The third argument is an untyped pointer to memory.

Input: <d> opens device node, returns file handle  
<request> argument in or out

The interface is shown as below.

```
int ioctl(int d, unsigned long request,...);
```

Refer to desktop Linux's man page for detailed documentation:

```
moxa@Moxa:~# man ioctl
```

## RTC (Real Time Clock)

The device node for the RTC is located at **/dev/rtc**. The V2101 supports standard Linux RTC controls. To access the RTC in a script you must **include <linux/rtc.h>**.

1. Function: **RTC\_RD\_TIME**

Description: reads the time from RTC. It will return the value on argument 3.

```
int ioctl(fd, RTC_RD_TIME, struct rtc_time *time);
```

2. Function: **RTC\_SET\_TIME**

Description: set RTC time. Argument 3 will be passed to RTC.

```
int ioctl(fd, RTC_SET_TIME, struct rtc_time *time);
```

## UART

The normal tty device nodes for the UART are **/dev/ttyS0** and **/dev/ttyS1**.

The V2101 supports standard Linux **termios** control with RS-232/422/485 serial ports.

To configure the serial ports, follow these steps.

1. To access the UART in a script you should call the **moxadevice.h** software library, which you can find in the folder **\example\moxalib** on your software CD.

```
#define RS232_MODE 0
#define RS485_2WIRE_MODE 1
#define RS422_MODE 2
#define RS485_4WIRE_MODE 3
```

2. The function **MOXA\_SET\_OP\_MODE** sets the interface mode. The function takes three arguments, and the third will set the UART device driver to the desired interface.

```
int ioctl(fd, MOXA_SET_OP_MODE, &mode)
```

3. The function **MOXA\_GET\_OP\_MODE** will return the current interface mode as the third argument.

```
int ioctl(fd, MOXA_GET_OP_MODE, &mode)
```

There are two Moxa private **ioctl** definitions for setting up special baudrates. If you use this **ioctl** to set a special baudrate, the **termios** cflag will be **B4000000**, in which case the **B4000000** definition will change. If the baudrate you get from **termios** (or from calling **tcgetattr()**) is **B4000000**, you must call **ioctl** with **MOXA\_GET\_SPECIAL\_BAUD\_RATE** to get the actual baudrate.

```
MOXA_SET_SPECIAL_BAUD_RATE
MOXA_GET_SPECIAL_BAUD_RATE
```

## Sample Script for Setting the Baudrate

```
#include "moxadevice.h"
#include <termios.h>
struct termios term;
int fd, speed;
fd = open("/dev/ttyS0", O_RDWR);
tcgetattr(fd, &term);
term.c_cflag &= ~(CBAUD | CBAUDEX);
term.c_cflag |= B4000000;
tcsetattr(fd, TCSANOW, &term);
speed = 500000;
ioctl(fd, MOXA_SET_SPECIAL_BAUD_RATE, &speed);
```

## Sample Script for Returning the Baudrate

```
#include "moxadevice.h"
#include <termios.h>
struct termios term;
int fd, speed;
fd = open("/dev/ttyS0", O_RDWR);
tcgetattr(fd, &term);
if ( (term.c_cflag & (CBAUD|CBAUDEX)) != B4000000 ) {
// follow the standard termios baud rate define
} else {
ioctl(fd, MOXA_GET_SPECIAL_BAUD_RATE, &speed);
}
```

## Non-Standard Baudrates and Inaccuracy

Moxa's UART ASIC features two registers: a standard 16 bit register that supports all standard baudrates, and a secondary register that adds support for non-standard baudrates in the range 50 bps to 921.6 Kbps.

Because serial devices which use non-standard baudrates generally use slower baudrates, supported baudrates are much denser in the lower range than in the upper (i.e., no baudrates are supported between the integers 5320 and 5323, but 49 baudrates are supported between the integers 387 and 388).

However, before using a serial device that uses a non-standard baudrate, system engineers should first check that this computer is compatible with the target baudrate within the accuracy tolerance specified by the serial device manufacturer.

Use formula B (shown below) to calculate the closest baudrate that can be achieved for any given non-standard baudrate; if this result falls within the rate tolerance supported by the device, then the computer is compatible with the target device. In the equations below:

- **Baudrate** represents the desired target baudrate
- **N** represents the multiple to which the standard UART register is set
- **M** represents the value for the secondary register that Moxa devices use to adjust transmission rates for non-standard devices; M is a number between 0 (representing standard, non-fractional baudrate multiplier) and 7. In the Moxa UART API, M is represented as **ENUM**.

As shown in the equations below, Moxa's secondary register can only be set to intervals of sevenths (e.g, 0.125, 0.250, 0.375, etc.). It is because of this limitation that engineers need to check first and see if the UART registers are compatible within a device's specified transmission tolerances.

**To calculate a standard baudrate, use formula A:**

(A) **Baudrate** =  $8 \times 921600/N$  bps, where:

**N** is the multiplier for the standard UART register; here, it is any integer between 1 (912,600 bps) and 18432 (50 bps).

**To calculate non-standard baudrates, use formula B:**

(B) **Baudrate** =  $921600 \div (N + (M/8))$  bps, where:

**N** is a the standard register divisor; this will be an integer between 1 and 18,432.

**M** is an integer between 0 and 7; the fraction  $M/8$  will calculate the fractional rate to which the secondary register will be adjusted to compensate for non-standard baudrates.

**Sample Code Showing a Typical UART Configuration Script:**

```
#define SET_MOXA_MUST_ENUM_VALUE(baseio, Value) { \
    UCHAR    __oldlcr, __efr; \
    __oldlcr = inb((baseio)+UART_LCR); \
    outb(MOXA_MUST_ENTER_ENCHANCE, (baseio)+UART_LCR); \
    __efr = inb((baseio)+MOXA_MUST_EFR_REGISTER); \
    __efr &= ~MOXA_MUST_EFR_BANK_MASK; \
    __efr |= MOXA_MUST_EFR_BANK2; \
    outb(__efr, (baseio)+MOXA_MUST_EFR_REGISTER); \
    outb((UCHAR) Value, (baseio)+MOXA_MUST_ENUM_REGISTER); \
    outb(__oldlcr, (baseio)+UART_LCR); \
}

quot = 921600 / 100000; // here 100000 is want to set baud rate, 921600 is a constant
which is depended on hardware
outb(cval | UART_LCR_DLAB, info->base + UART_LCR); /* set DLAB */
outb(quot & 0xff, info->base + UART_DLL); /* LS of divisor */
outb(quot >> 8, info->base + UART_DLM); /* MS of divisor */
outb(cval, info->base + UART_LCR); /* reset DLAB */

quot = 921600 % 100000;
quot *= 8;
if ( (quot % 100000) > (100000 / 2) ) {
    quot /= 100000;
    quot++;
} else {
    quot /= 100000;
}

SET_MOXA_MUST_ENUM_VALUE(info->base, quot);
```

**Example:** Your serial device requires using a baudrate of 5340 bps and has a transmission tolerance of  $\pm 2$  bps. Can this computer be used with this device?

**Solution:** Set formula B to the desired baudrate and then solve for M.

$$5338 = 8 \times 921600 / K \implies M = 1367.703259\dots$$

This shows that the supported baudrate closest to 5340 comes from setting K to 1367 or 1368.

$$M=1368 \implies \text{Baudrate1} = 5336.842105\dots$$

$$M=1367 \implies \text{Baudrate2} = 5340.746159\dots$$

Because  $(5338 - \text{Baudrate1}) < 2$ , this computer will transmit at this non-standard rate (Baudrate 1) within the accuracy tolerance specified by the device.

Note that we can also use formula A to generate the so-called "standard" baudrates, which come from setting  $M=0$ , and setting N equal to certain integers.

| Standard Baudrates |     |   |          |       |   |
|--------------------|-----|---|----------|-------|---|
| Baudrate           | N   | M | Baudrate | N     | M |
| 921600             | 1   | 0 | 4800     | 192   | 0 |
| 460800             | 2   | 0 | 2400     | 384   | 0 |
| 230400             | 4   | 0 | 1800     | 512   | 0 |
| 115200             | 8   | 0 | 1200     | 768   | 0 |
| 57600              | 16  | 0 | 600      | 1536  | 0 |
| 38400              | 24  | 0 | 300      | 3072  | 0 |
| 19200              | 48  | 0 | 150      | 6144  | 0 |
| 9600               | 96  | 0 | 75       | 12288 | 0 |
| 7200               | 128 | 0 | 50       | 18432 | 0 |



### WARNING

Communication between a serial device and a Moxa UART port may not work correctly if the serial device communicates at a baud rate that is not within the tolerance of a rate calculated from either formula A or formula B.

### Special Note

1. The termios **cflag** will only return baudrates from the first register (i.e.: standard baudrates like 50, 75, 110, 2400, 4800, 115200, 230400, 921600, and so forth). If the UART register is configured for a non-standard baudrate, the termios cflag will **not** reflect the actual baud rate.
2. Because of the above limitation, if administrators use a serial terminal (**stty**) to retrieve serial information about non-standard baudrates, the terminal will return a speed of **0** for any non-standard baudrates.

## Digital I/O

Digital output channels can be set to high or low. The channels are controlled by the function call **set\_dout\_state( )**. Use the digital input channels to detect the state change of the digital input signal. The DI channels can also be used to detect whether or not the state of a digital signal changes during a fixed period of time. This can be done by the function call, **set\_din\_event( )**.

#### Return error code definitions:

```
#define DIO_ERROR_PORT -1 // no such port
#define DIO_ERROR_MODE -2 // no such mode or state
#define DIO_ERROR_CONTROL -3 // open or ioctl fail
#define DIO_ERROR_DURATION -4 // The value of duration is not 0 or not in the range,
40 <= duration <= 3600000 milliseconds (1 hour)
#define DIO_ERROR_DURATION_20MS -5 // The value of duration must be a multiple of
20 ms
#define DIO_OK 0
```

#### DIN and DOUT definitions:

```
#define DIO_HIGH 1
#define DIO_LOW 0
```

#### Moxa functions for DI/DO

|             |                                                                                                                                               |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Function    | <b>int set_dout_state(int doport, int state)</b>                                                                                              |
| Description | Set the DOUT port to high or low state.                                                                                                       |
| Input       | <doport> The DOUT port you want to set. Port starts from 0 to 3<br><state> Set high or low state; DIO_HIGH (1) for high, DIO_LOW (0) for low. |

|        |                         |
|--------|-------------------------|
| Output | none                    |
| Return | refer to the error code |

|             |                                                                                                            |
|-------------|------------------------------------------------------------------------------------------------------------|
| Function    | <b>int get_din_state(int diport, int *state)</b>                                                           |
| Description | Get the DIN port state                                                                                     |
| Input       | <diport> The DIN port to get the state of. Port numbering is from 0 to 3<br><state> Save the current state |
| Output      | <state> DIO_HIGH (1) for high, DIO_LOW (0) for low                                                         |
| Return      | Refer to the error code                                                                                    |

|             |                                                                                |
|-------------|--------------------------------------------------------------------------------|
| Function    | <b>int get_dout_state(int doport, int *state)</b>                              |
| Description | Get the DOUT port state                                                        |
| Input       | <doport> The DOUT port to get the state of.<br><state> Save the current state. |
| Output      | <state> DIO_HIGH (1) for high, DIO_LOW (0) for low                             |
| Return      | Refer to the error code                                                        |

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Function    | <b>int set_din_event(int diport, void (*func)(int diport), int mode, long int duration)</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Description | Set the DIN event when the state is changed from high to low or from low to high.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Input       | <diport> The port that will be used to detect the DIN event.<br>Port numbering is from 0 to 3. This value depends on your device.<br><(*func) (int diport)><br>Not NULL: Returns the call back function. When the event occurs, the call back function will be invoked.<br>NULL: Clear this event<br><mode><br>DIN_EVENT_HIGH_TO_LOW (1): From high to low<br>DIN_EVENT_LOW_TO_HIGH (0): From low to high<br>DIN_EVENT_CLEAR (-1): Clear this event<br><duration><br>0: Detect the din event DIN_EVENT_HIGH_TO_LOW or DIN_EVENT_LOW_TO_HIGH without duration<br>Not 0: Detect the din event DIN_EVENT_HIGH_TO_LOW or DIN_EVENT_LOW_TO_HIGH with duration.<br>Note:<br>The value of "duration" must be a multiple of 20 milliseconds.<br>The range of "duration" is 0, or 40 <= duration <= 3600000 milliseconds.<br>The error of the measurement is 24 ms. For example, if the DIN duration is 200 ms, this event will be generated when the DIN pin stays in the same state for a time between 176 ms and 200 ms. |
| Output      | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Return      | Refer to the error code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

|             |                                                                                                                                                                                                   |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Function    | <b>int get_din_event(int diport, int *mode, long int *duration)</b>                                                                                                                               |
| Description | To retrieve the DIN event configuration, including mode (DIN_EVENT_HIGH_TO_LOW or DIN_EVENT_LOW_TO_HIGH), and the value of "duration."                                                            |
| Input       | <diport> Which DIN port you want to retrieve<br><mode> Save the set event.<br><duration> The duration the DIN port is kept in high or low state. - return to the current duration value of diport |
| Output      | <mode><br>DIN_EVENT_HIGH_TO_LOW (1): From high to low<br>DIN_EVENT_LOW_TO_HIGH(0): From low to high                                                                                               |

|        |                                                                                                                                     |
|--------|-------------------------------------------------------------------------------------------------------------------------------------|
|        | DIN_EVENT_CLEAR(-1): Clear this event<br><duration><br>The value of duration should be 0 or 40 <= duration <= 3600000 milliseconds. |
| Return | Refer to the error code                                                                                                             |

## Special Note

1. The Moxa code library, **moxalib**, must first be compiled before the digital input and output channels can be enabled. The Moxa library may be found on the software CD included with your computer, in the folder `/media/cdrom/example/moxalib` (after mounting the CD in your computer). To build **moxalib**, call the GNU program **make** from the home directory, as an unprivileged user: this will protect the system from unanticipated corruption that might be introduced during the build process if one were to run it with full root privileges. In order to compile the library from your home drive you will need to first copy it over from the software CD. Thus, to compile **moxalib** you will need to run the following commands:

```
moxa@Moxa:~# cd /home/
moxa@Moxa:~/home# mkdir code && cd code
moxa@Moxa:~/home/code# cp -a /media/cdrom/example/moxalib .
moxa@Moxa:~/home/code# cd moxalib
moxa@Moxa:~/home/code/moxalib# make all
```



### WARNING

**Do not run the make command using root privileges.** Doing so could harm your system, and will eliminate all of the system protections that a user account provides against buggy code and other problems. Additionally, remember that **every time you update the kernel you will need to rebuild this library.** If or when you upgrade your kernel to a new version you will need to recompile this library, along with any other kernel modules for third party software you may have installed.

2. If you want to build the DIO sample code, follow this command:  
**moxa@Moxa:~/home/code/dio# make**  
Please note that the working directory in this example is `/home/code/dio`.
3. Make sure to link the library **libmoxalib** for DI/DO programming, and include the header file **moxadevice.h**. Only one program at a time can use the DI/DO library.
4. Due to hardware limitation, you need to modify `MIN_DURATION` as 60 for V2101.

## Sample Code for Digital I/O Control

File Name: `tdio.c`

Description: This program connects Digital Output 1 to Digital Input 1, changes the digital output state to high or low according to manual input, then detects and counts the state changes from Digital Input 1.

```
#include <stdio.h>
#include <stdlib.h>
#ifdef NO_MOXADEVICE_HEADER
#include "moxadevice.h"
#else
#include <moxadevice.h>
#endif
#include <fcntl.h>
/* Due to hardware limitation, MIN_DURATION should be 60 for DA710 */
#define MIN_DURATION 40
static char *DataString[2]={"Low ", "High "};
```



```

static void hightolowevent(int diport)
{
    printf("\nDIN port %d high to low.\n", diport);
}
static void lowtohighevent(int diport)
{
    printf("\nDIN port %d low to high.\n", diport);
}
int main(int argc, char * argv[])
{
    int    i, j, state, retval;
    unsigned long duration;
    while( 1 ) {
        printf("\nSelect a number of menu, other key to exit.  \n\
1.set high to low event      \n\
2.get now data.              \n\
3.set low to high event     \n\
4.clear event                \n\
5.set high data.            \n\
6.set low data.             \n\
7. quit                      \n\
8. show event and duration   \n\
Choose : ");
        retval =0;
        scanf("%d", &i);
        if ( i == 1 ) { // set high to low event
            printf("Please keyin the DIN number : ");
            scanf("%d", &i);
            printf("Please input the DIN duration, this minimun value must be over
%d : ",MIN_DURATION);
            scanf("%lu", &duration);
            retval=set_din_event(i, hightolowevent, DIN_EVENT_HIGH_TO_LOW,
duration);
        } else if ( i == 2 ) { // get now data
            printf("DIN data : ");
            for ( j=0; j<MAX_DIN_PORT; j++ ) {
                get_din_state(j, &state);
                printf("%s", DataString[state]);
            }
            printf("\n");
            printf("DOUT data : ");
            for ( j=0; j<MAX_DOUT_PORT; j++ ) {
                get_dout_state(j, &state);
                printf("%s", DataString[state]);
            }
            printf("\n");
        } else if ( i == 3 ) { // set low to high event
            printf("Please keyin the DIN number : ");
            scanf("%d", &i);
            printf("Please input the DIN duration, this minimun value must be over
%d : ",MIN_DURATION);
            scanf("%lu", &duration);
            retval = set_din_event(i, lowtohighevent, DIN_EVENT_LOW_TO_HIGH,
duration);
        } else if ( i == 4 ) { // clear event

```

```

        printf("Please keyin the DIN number : ");
        scanf("%d", &i);
        retval=set_din_event(i, NULL, DIN_EVENT_CLEAR, 0);
    } else if ( i == 5 ) { // set high data
        printf("Please keyin the DOUT number : ");
        scanf("%d", &i);
        retval=set_dout_state(i, 1);
    } else if ( i == 6 ) { // set low data
        printf("Please keyin the DOUT number : ");
        scanf("%d", &i);
        retval=set_dout_state(i, 0);
    } else if ( i == 7 ) { // quit
        break;
    } else if ( i == 8 ) { // show event and duration
        printf("Event:\n");
        for ( j=0; j<MAX_DOUT_PORT; j++ ) {
            retval=get_din_event(j, &i, &duration);
            switch ( i ) {
                case DIN_EVENT_HIGH_TO_LOW :
                    printf(" (htl,%lu) ", duration);
                    break;
                case DIN_EVENT_LOW_TO_HIGH :
                    printf(" (lth,%lu) ", duration);
                    break;
                case DIN_EVENT_CLEAR :
                    printf(" (clr,%lu) ", duration);
                    break;
                default :
                    printf("err " );
                    break;
            }
        }
        printf("\n");
    } else {
        printf("Select error, please select again !\n");
    }
}
switch(retval) {
    case DIO_ERROR_PORT:
        printf("DIO error port\n");
        break;
    case DIO_ERROR_MODE:
        printf("DIO error mode\n");
        break;
    case DIO_ERROR_CONTROL:
        printf("DIO error control\n");
        break;
    case DIO_ERROR_DURATION:
        printf("DIO error duratoin\n");
    case DIO_ERROR_DURATION_20MS:
        printf("DIO error! The duratoin is not a multiple of 20 ms\n");
        break;
}
}
return 0;
}

```

```
DIO Program Make File Example
include ../compile.mk
CC=$(PREFIX)gcc
STRIP=$(PREFIX)strip
AR=$(PREFIX)ar
LNAME=moxalib
all:    release
release: $(MOXALIB_OBJ)
        $(AR) rcs lib$(LNAME).a $(MOXALIB_OBJ)
%.o:%.c
        $(CC) -c $<
install:    lib$(LNAME).a
        cp -a lib$(LNAME).a $(MOXALIB_INSTALL_DIR)
        cp -a moxadevice.h /usr/local/arm-linux/include
        cp -a moxadevice.h /usr/local/arm-linux/arm-linux/include
clean:
        /bin/rm -f *.o *.a
```

## Built-In Linux OS Recovery

---

The V2101-LX is installed with the Embedded Linux operating system, which is located in the Flash DOM (CompactFlash card) shipped with the V2101-LX computer. Although it rarely happens, you may find on occasion that operating system files and/or the disk file system are damaged. This chapter describes how to recover the Linux operating system.

The following topics are covered in this chapter:

□ **Recovery Environment**

□ **Recovery Procedure**

- Step 1: Format an Empty USB Disk.
- Step 2: Create a Linux Bootable USB Disk.
- Step 3: Set up the BIOS to Boot from a USB Disk.
- Step 4: Recover the Linux system from a USB Disk.

# Recovery Environment

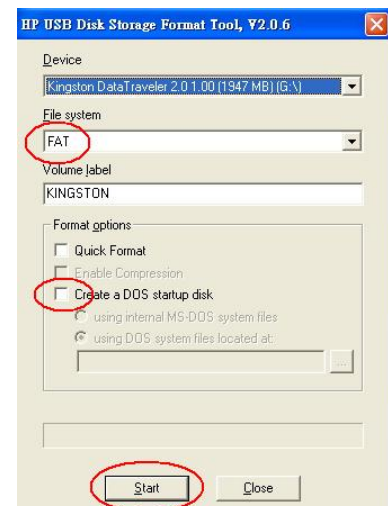
The recovery environment includes the V2101-LX embedded computer and a bootable USB disk with the recovery programs and system image file.



## Recovery Procedure

### Step 1: Format an Empty USB Disk.

- Prepare a USB disk that has at least a 256 MB capacity.
- Format your USB disk with the **HP USB Disk Format Tool**. Open the utility and select the device and FAT file system. You need empty disk only. **DO NOT** check the option **Create a DOS startup disk**.
- Click **Start**.

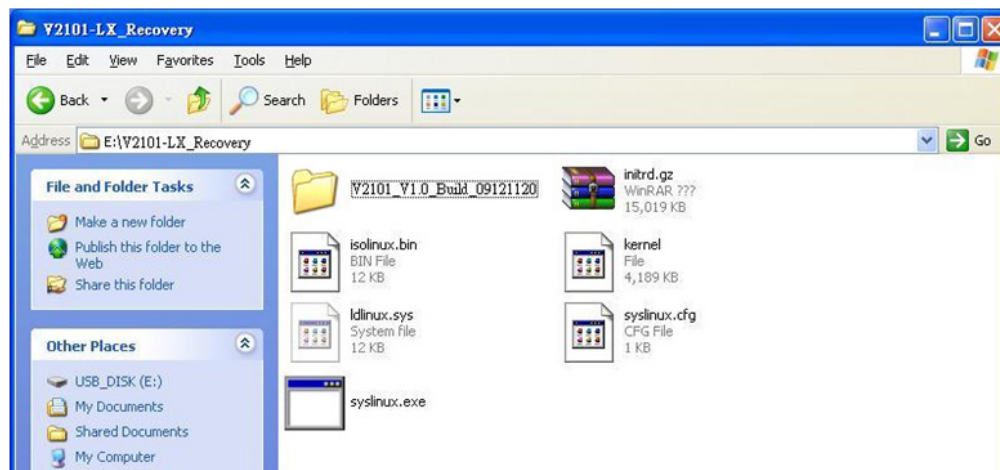


#### ATTENTION

The HP USB Disk Storage Format Tool can be downloaded from many web sites. Do a search on **HP USB Disk Storage Format Tool** from any search engine to locate the tool, or you may try the link below:  
<http://hp-usb-disk-storage-format-tool.soft32.com/free-download/>

### Step 2: Create a Linux Bootable USB Disk.

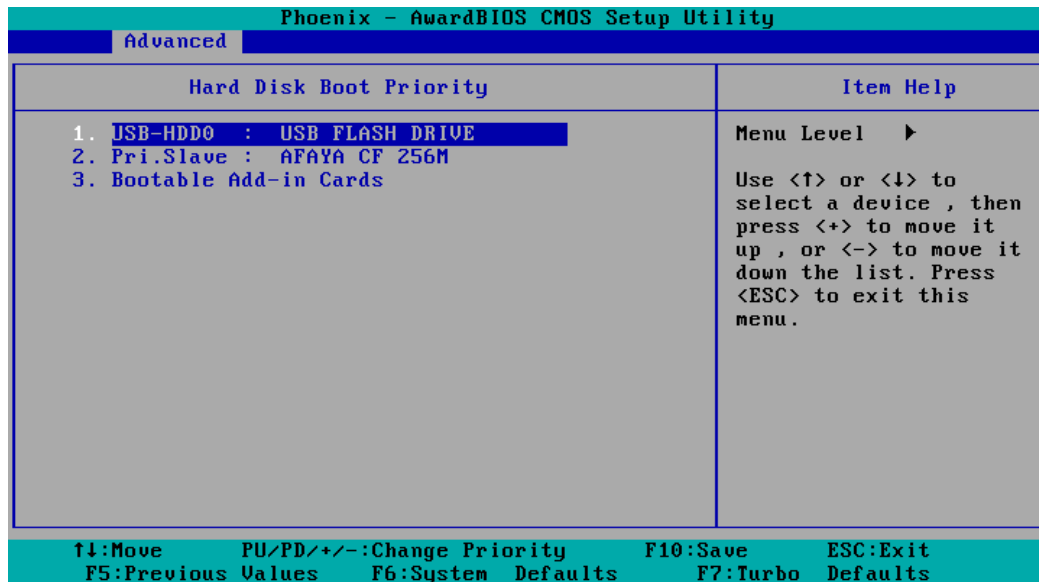
- You can find the **firmware** directory in the Recovery CD shipped with the V2101-LX computer.
- Configure Windows Explorer to show hidden files (including protected operating system files).
- Copy all files in the **firmware** directory to the root directory of your USB disk.



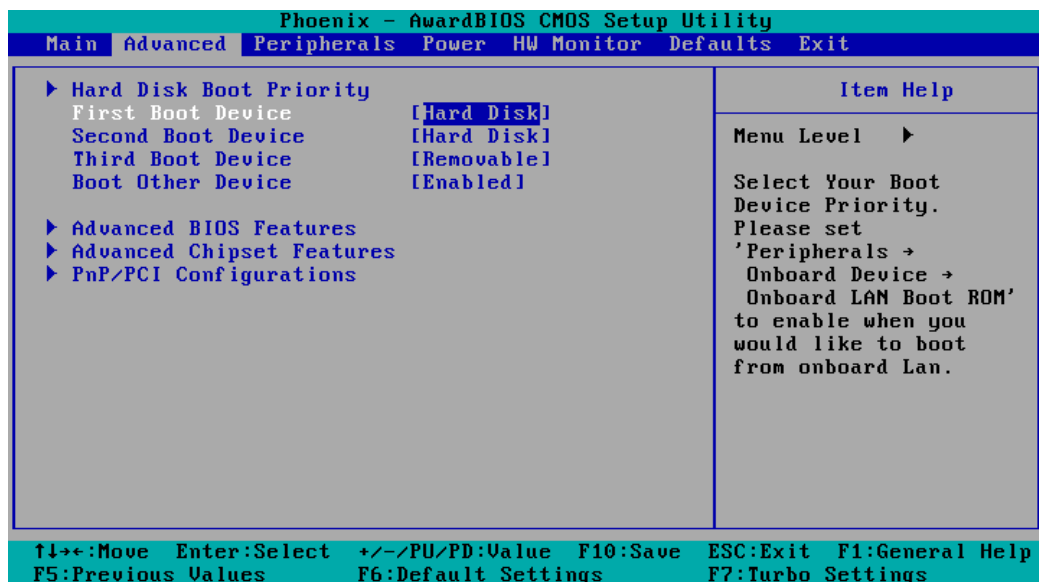
- d. Open a DOS prompt and type **M:\syslinux.exe M:** to create a bootable Linux disk.  
In this example, M: is the USB Disk drive number.

### Step 3: Set up the BIOS to Boot from a USB Disk.

- Insert the USB disk.
- Power on and press **DEL** to enter the bios setup menu.
- Select **Advanced** → **Hard Disk Boot Priority** and then press **Enter**.
- From the setup menu, use “↑” or “↓” to select the USB device



- Press “+” to move the selection up to the first priority, and press **Esc** to exit the setup menu.
- Make sure the first boot device is **Hard Disk**. If not, press **Enter** to change it.



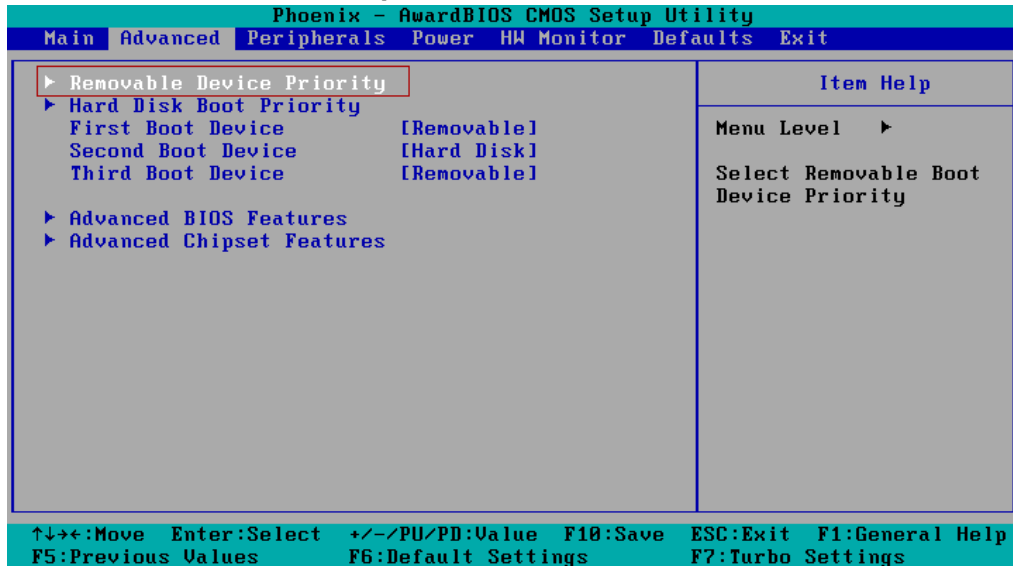
- Select **Exit** → **Save & Exit Setup** and then press **Enter**.
- Choose **Y** to save to the CMOS and then exit.



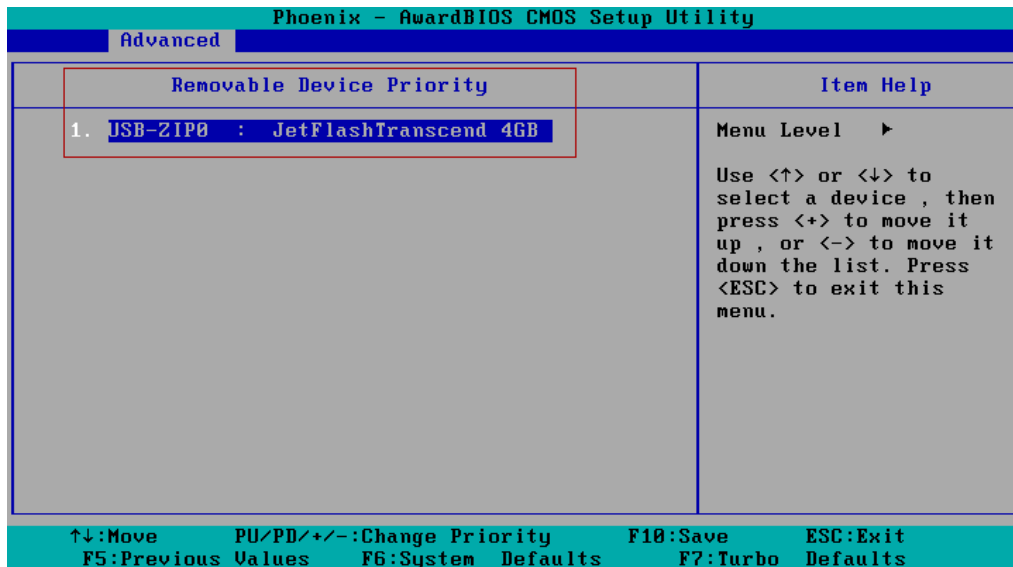
#### ATTENTION

Please note that some USB disks will be regarded as the **Removable Device**. If it happens, see the following steps.

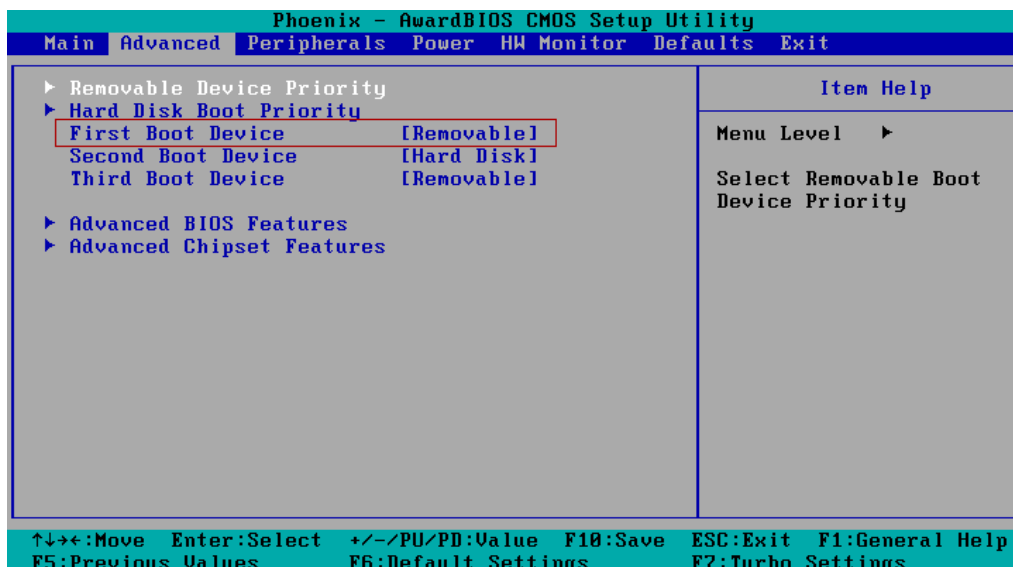
i. Select **Removable Device Priority**.



j. Make sure that the USB disk has been detected. Press **Esc** to exit.



k. Make sure that the **First Boot Device is Removable**. If not, select **First Boot Device**, press **Enter** and select it from the list.



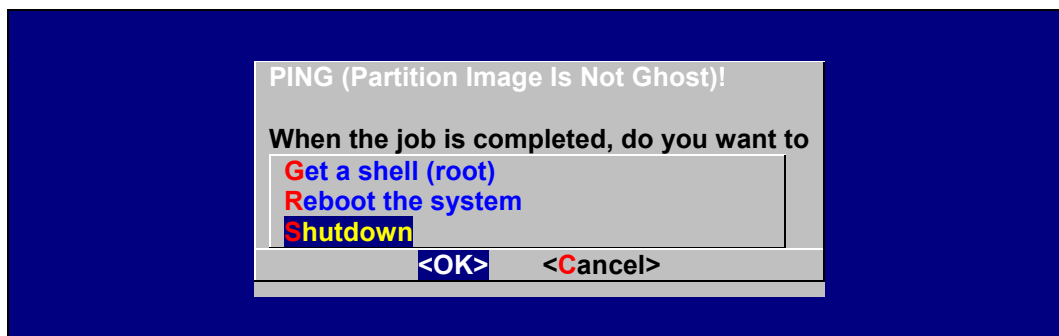
- l. Select **Exit** → **Save & Exit Setup** and then press **Enter**.
- m. Choose **Y** to save to the CMOS and then exit.

## Step 4: Recover the Linux system from a USB Disk.

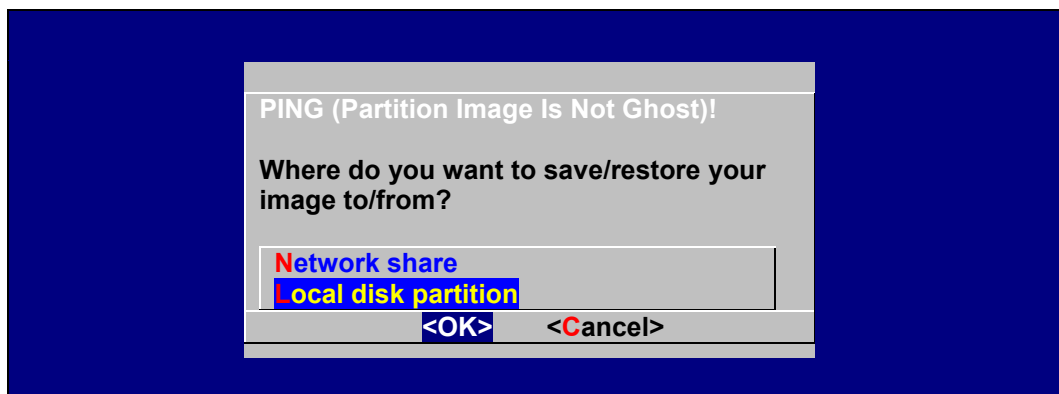
- a. If the BIOS setup is correct, it will boot from the USB disk. Follow the steps below to set up recovery parameters.



- b. Choose **OK** to go to the next step.
- c. Choose shut down the V2101-LX when the restoration is finished.

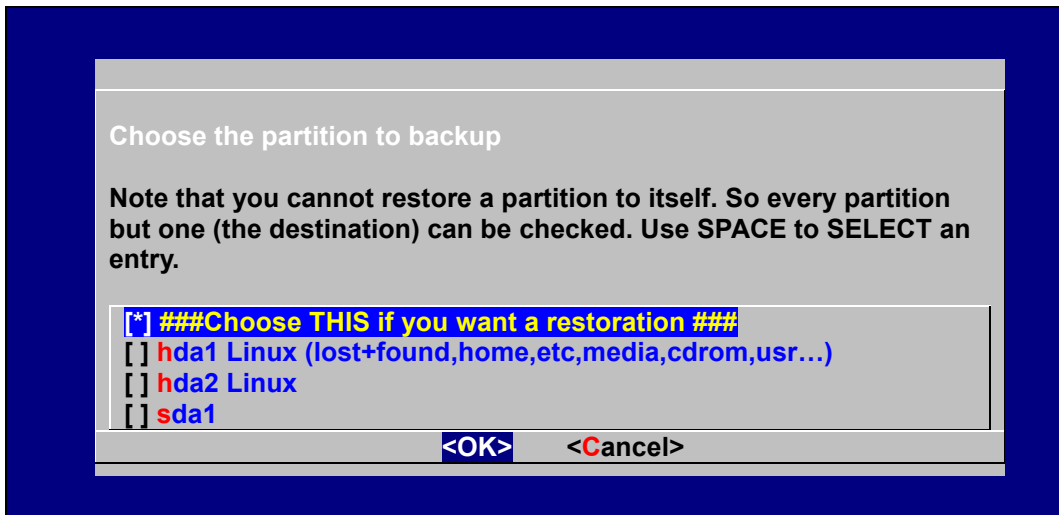


- d. Choose restore image from **Local disk partition**.

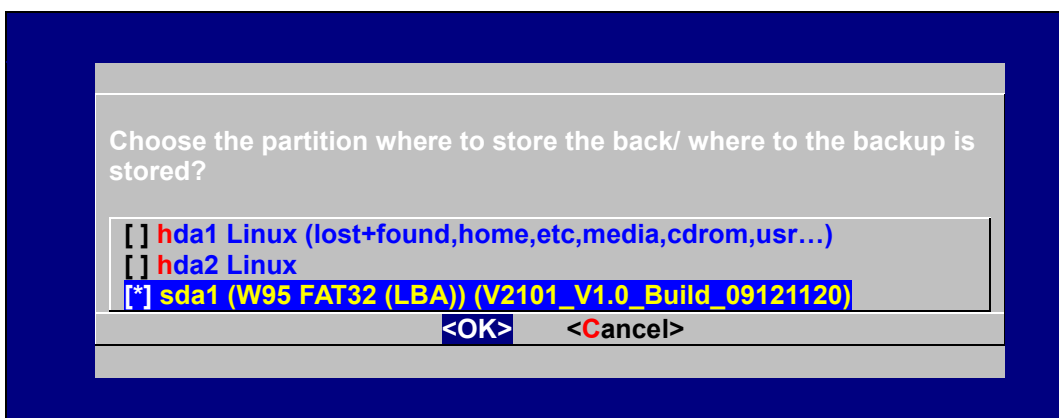




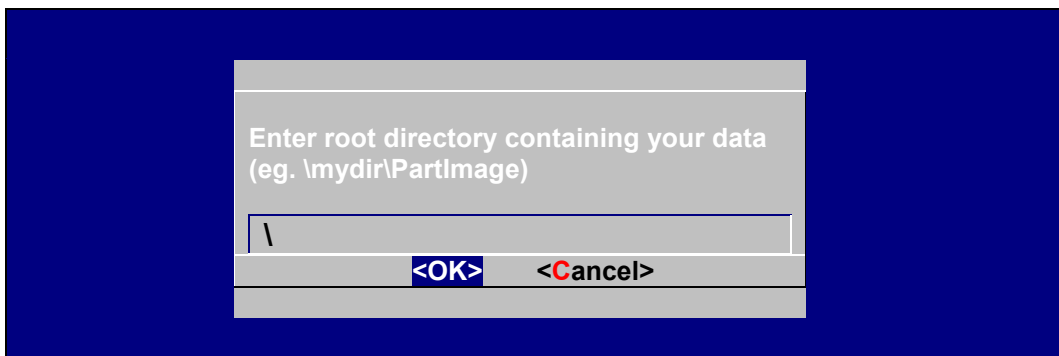
- e. Choose **### Choose THIS if you want a restoration ###**



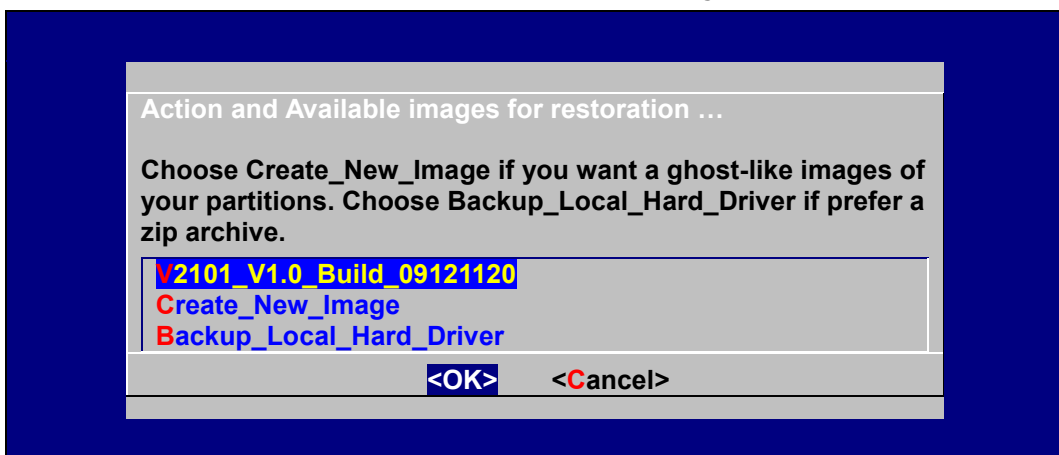
- f. Choose the restoration source device **sda1**.



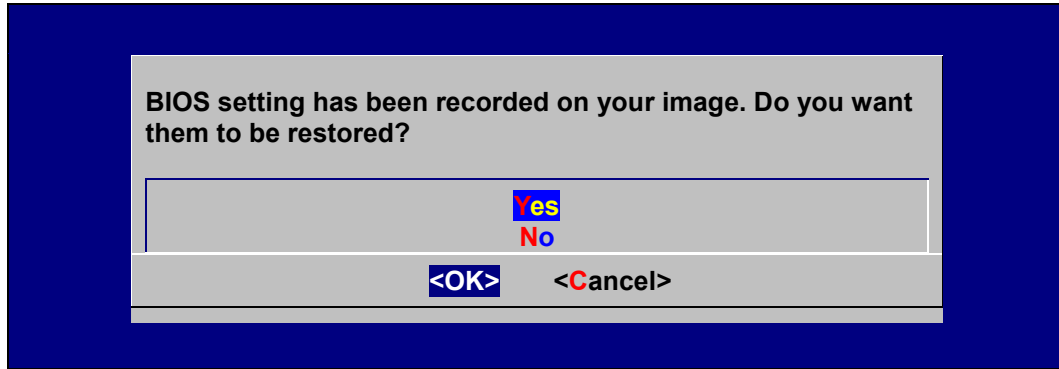
- g. Enter **"\"** to choose the root directory of the restoration image.



- h. Choose **V2101\_V1.0\_Build\_09121120** for the restoration image.



- i. Choose **Yes** to start the restoration. After the restoration is finished, the system will halt and you will need to reboot to restart the restored system.



When operation is finished, turn off the computer and remove the USB disk.

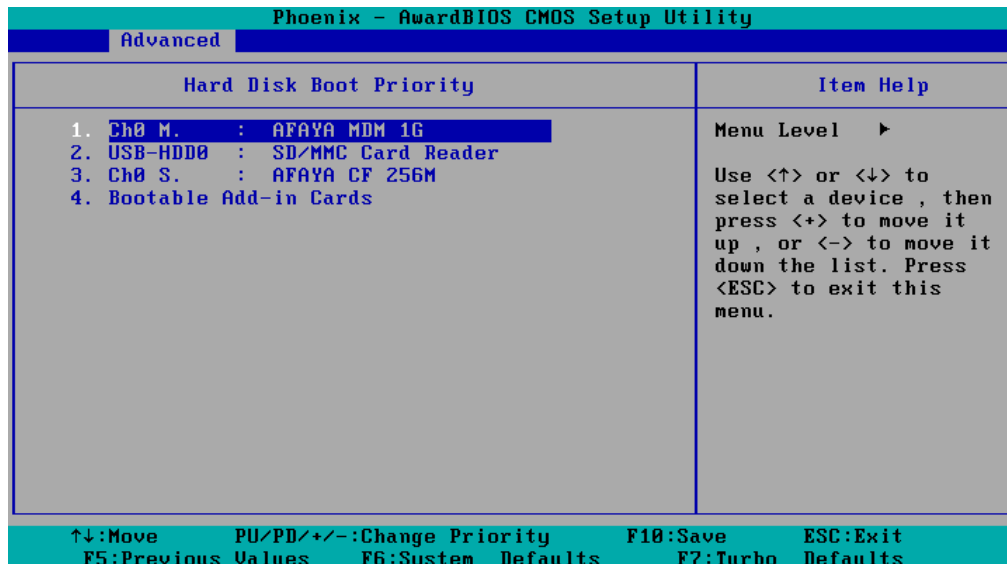


### ATTENTION

**DO NOT** turn off the power during system recovery, as the system may crash.

## Step 5: Reset the BIOS to boot from DOM or CompactFlash.

- Power on and press **DEL** to enter the bios setup menu.
- Select **Advanced** → **Hard Disk Boot Priority** and then press **Enter**.
- From the setup menu, use "↑" or "↓" to select the DOM or CompactFlash device.
- Press "+" to move the selection up to the first priority, and press **Esc** to exit the setup menu.
- Select **Exit** → **Save & Exit Setup** and then press **Enter**.
- Choose **Y** to save to the CMOS and then exit.
- Wait a few minutes for the system to boot. When the recovery process is finished, you will again be able to see the Linux desktop.



# A

## Sample Scripts & Firewalls

---

In this section, we present a basic firewall for your reference.

The following topics are covered in this appendix:

- ❑ **A Sample Initialization Script**
- ❑ **A Sample Firewall**

# A Sample Initialization Script

```

#!/bin/sh
# Copyright (c) XXXX <<Your Name Here>>
# All rights reserved.
#
#
# /etc/init.d/<<name of your script here>>
# and its symbolic link
# /usr/sbin/rc<<name of your script here>>

### BEGIN INIT INFO
# Provides:          <<name of your script here>>
# Required-Start:    $network
# Required-Stop:
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Short-Description: The <<name of your script here>> daemon provides...
# Description:       The <<name of your script here>> daemon is ...
#                    that is active in runlevels 3 and 5.
#
### END INIT INFO

# Check for missing binaries
<<NAME OF YOUR SCRIPT HERE>>_BIN=/usr/bin/<<name of your script here>>
test -x $<<NAME OF YOUR SCRIPT HERE>>_BIN || { echo "$<<NAME OF YOUR SCRIPT HERE>>_BIN
not installed";
    if [ "$1" = "stop" ]; then exit 0;
    else exit 5; fi; }

# Check for existence of needed config file and read it
<<NAME OF YOUR SCRIPT HERE>>_CONFIG=/etc/<<name of your script here>>.cfg
test -r $<<NAME OF YOUR SCRIPT HERE>>_CONFIG || { echo "$<<NAME OF YOUR SCRIPT HERE>>_CONFIG
not existing";
    if [ "$1" = "stop" ]; then exit 0;
    else exit 6; fi; }

# Read config
. $<<NAME OF YOUR SCRIPT HERE>>_CONFIG

# Load the rc.status script for this service.
. /etc/rc.status

# Reset status of this service
rc_reset

case "$1" in
start)
    echo -n "Starting <<name of your script here>> "
    ## Start daemon with startproc(8). If this fails
    ## the return value is set appropriately by startproc.
    startproc $<<NAME OF YOUR SCRIPT HERE>>_BIN

    # Remember status and be verbose

```

```

    rc_status -v
    ;;
stop)
    echo -n "Shutting down <<name of your script here>> "
    ## Stop daemon with killproc(8) and if this fails
    ## killproc sets the return value according to LSB.

    killproc -TERM $<<NAME OF YOUR SCRIPT HERE>>_BIN

    # Remember status and be verbose
    rc_status -v
    ;;
restart)
    ## Stop the service and regardless of whether it was
    ## running or not, start it again.
    $0 stop
    $0 start

    # Remember status and be quiet
    rc_status
    ;;
reload)
    # If it supports signaling:
    echo -n "Reload service bar "
    killproc -HUP $BAR_BIN
    #touch /var/run/<<NAME OF YOUR SCRIPT HERE>>.pid
    rc_status -v

    ## Otherwise if it does not support reload:
    #rc_failed 3
    #rc_status -v
    ;;
status)
    echo -n "Checking for service <<name of your script here>> "
    ## Check status with checkproc(8), if process is running
    ## checkproc will return with exit status 0.

    # Return value is slightly different for the status command:
    # 0 - service up and running
    # 1 - service dead, but /var/run/ pid file exists
    # 2 - service dead, but /var/lock/ lock file exists
    # 3 - service not running (unused)
    # 4 - service status unknown :-(
    # 5--199 reserved (5--99 LSB, 100--149 distro, 150--199 appl.)

    # NOTE: checkproc returns LSB compliant status values.
    checkproc $<<NAME OF YOUR SCRIPT HERE>>_BIN
    # NOTE: rc_status knows that we called this init script with
    # "status" option and adapts its messages accordingly.
    rc_status -v
    ;;
*)
    ## If no parameters are given, print which are available.
    echo "Usage: $0 {start|stop|status|restart|reload}"

```

```
    exit 1
;;
esac
rc_exit
```

## A Sample Firewall

```
#!/bin/bash
# If you put this shell script in the /home/nat.sh
# Remember to chmod 744 /home/nat.sh
# Edit the rc.local file to make this shell startup automatically.
# vi /etc/rc.local
# Add a line in the end of rc.local /home/nat.sh
EXIF="eth0" #This is an external interface for setting up a valid IP address.
EXNET="192.168.4.0/24" #This is an internal network address.
# Step 1. Insert modules.
# Here 2> /dev/null means the standard error messages will be dump to null device.
modprobe ip_tables 2> /dev/null
modprobe ip_nat_ftp 2> /dev/null
modprobe ip_nat_irc 2> /dev/null
modprobe ip_conntrack 2> /dev/null
modprobe ip_conntrack_ftp 2> /dev/null
modprobe ip_conntrack_irc 2> /dev/null
# Step 2. Define variables, enable routing and erase default rules.
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
export PATH
echo "1" > /proc/sys/net/ipv4/ip_forward
/sbin/iptables -F
/sbin/iptables -X
/sbin/iptables -Z
/sbin/iptables -F -t nat
/sbin/iptables -X -t nat
/sbin/iptables -Z -t nat
/sbin/iptables -P INPUT ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables -t nat -P PREROUTING ACCEPT
/sbin/iptables -t nat -P POSTROUTING ACCEPT
/sbin/iptables -t nat -P OUTPUT ACCEPT
# Step 3. Enable IP masquerade.
#ehco 1 > /proc/sys/net/ipv4/ip_forward
#modprobe ipt_MASQUERADE
#iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```