DA-820 Linux Software Manual

Version 1.0, December 2014

www.moxa.com/product

DA-820 Linux Software 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

Trademarks

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

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

1

Introduction

Thank you for purchasing the Moxa DA-820 Series of x86 ready-to-run embedded computers. This manual introduces the software configuration and management of the DA-820-LX computer, which runs the Linux operating system. For hardware installation, connector interfaces, setup, and upgrading the BIOS, refer to the *DA-820 Series Embedded Computer User's Manual.*

Linux is an open, scalable operating system that allows you to develop a wide range of innovative, small footprint devices. Software written for desktop PCs can be easily ported to the Linux-based embedded computer with a GNU 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

The DA-820 Series embedded computers are based on the Intel 3rd-generation processor and feature two serial ports, four 10/100/1000 Mbps LAN ports, six USB 2.0 hosts, and a CFast socket. The DA-820 Series offers dual VGA outputs, making it exceptionally well suited for industrial applications such as SCADA and factory automation.

The two serial ports on the DA-820 Series make it ideal for connecting a wide range of serial devices, and the four 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 DA-820 computers come with eight programmable LEDs for status indication. In addition, the CFast socket, and USB and SATA ports provide DA-820 computers with data buffering and storage expansion, which provide the necessary reliability for industrial applications. The IRIG-B time protocol is widely used by electric utilities, industrials, and others to ensure precise time synchronization of power system devices, such as breakers, relays, and meters. The DA-820 Series has an IRIG-B expansion module for your power system to support IRIG-B time synchronization.

Pre-installed with Linux, the DA-820 Series provides programmers with a friendly environment for developing sophisticated, bug-free application software at a lower cost.

Software Specifications

The Linux operating system pre-installed on the DA-820 embedded computers is the **Debian Wheezy** distribution. The Debian project involves a worldwide group of volunteers who endeavor to produce an operating system distribution 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.

The following figure shows an example software architecture.

ATTENTION

Depending on the Linux distribution and build version, not all components are included as shown in the figure.

ATTENTION

Refer to http://www.debian.org/ and http://www.gnu.org/ for information and documentation related to Debian GNU/Linux and open source software.

Software Components

The DA-820-LX computers are pre-installed with the Debian Wheezy 7.2 Linux distribution. For more information on the software components, see **Appendix A**.

2

Software Configuration

This chapter describes how to configure and manage a DA-820-LX computer directly from your desktop. There are two ways to connect to the DA-820-LX computer: through a VGA monitor or an SSH console. This chapter describes basic Linux operating system settings.

The following topics are covered in this chapter:

Accessing the DA-820

You can access the DA-820 using one of the following methods:

- A monitor connected to the DA-820.
- An SSH client from a computer.

Using a Monitor

1. Connect a monitor to the DA-820 and turn on the monitor and the DA-820.

NOTE The Da-820 may take up to 60 seconds to boot up.

2. A login screen appears. Enter the login information.

For more information, see the **Logging in for the First Time** and **Account Management** sections.

Using an SSH Client

You can access the DA-820 using an SSH client on a computer.

The DA-820-LX comes with four Gigabit ports (LAN1, LAN2, LAN3, and LAN4). The following table shows the default IP address and subnet mast settings on each port.

	Default IP Address	**Subnet Mask**
LAN 1	192.168.3.127	255.255.255.0
LAN 2	192.168.4.127	255.255.255.0
LAN 3	192.168.5.127	255.255.255.0
LAN 4	192.168.6.127	255.255.255.0

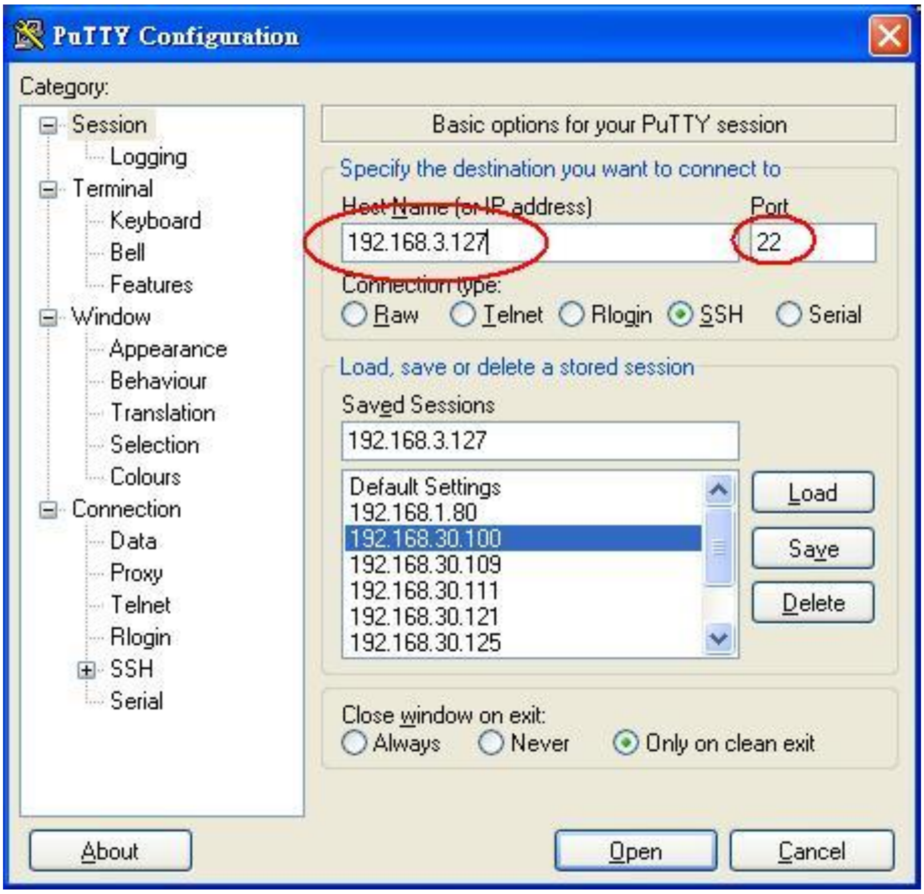
1. Use a cross-over Ethernet cable to connect your computer to one of the Ethernet ports on the DA-820.
2. Make sure that your computer is on the same subnet as the DA-820.
   For example, if your computer is connected to the LAN1 port on the DA-820, set the IP address of your computer to 192.168.3.126 and the subnet mask to 255.255.255.0. If your computer is connected to the LAN2 port on the DA-820, set the IP address of your computer to 192.168.4.126 and the subnet mask to 255.255.255.0.
3. Start the SSH client on your computer to establish a connection to the DA-820.
4. A login screen appears. Enter the login information.

For more information, see the **Logging in for the First Time** and **Account Management** sections.

Windows Users

This section shows you how to access the DA-820 from a Windows computer.

1. Start an SSH client application.
   You can install PuTTY (a free SSH client) in Windows.
2. Configure the connection information (for example, the host IP address and port number) to connect to the DA-820. The following figure shows the PuTTY configuration screen.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.3.127

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: 192.168.3.127

Default Settings
192.168.1.80
192.168.30.100
192.168.30.109
192.168.30.111
192.168.30.121
192.168.30.125

Close window on exit: ○ Always ○ Never ● Only on clean exit

Linux Users

1. To access the DA-820 from a Linux computer, use the ssh command to establish an SSH connection.
   For example: `ssh 192.168.3.127`
2. When prompted, enter “yes” to start the connection.

```
root@Moxa:~# 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_
```

Logging in for the First Time

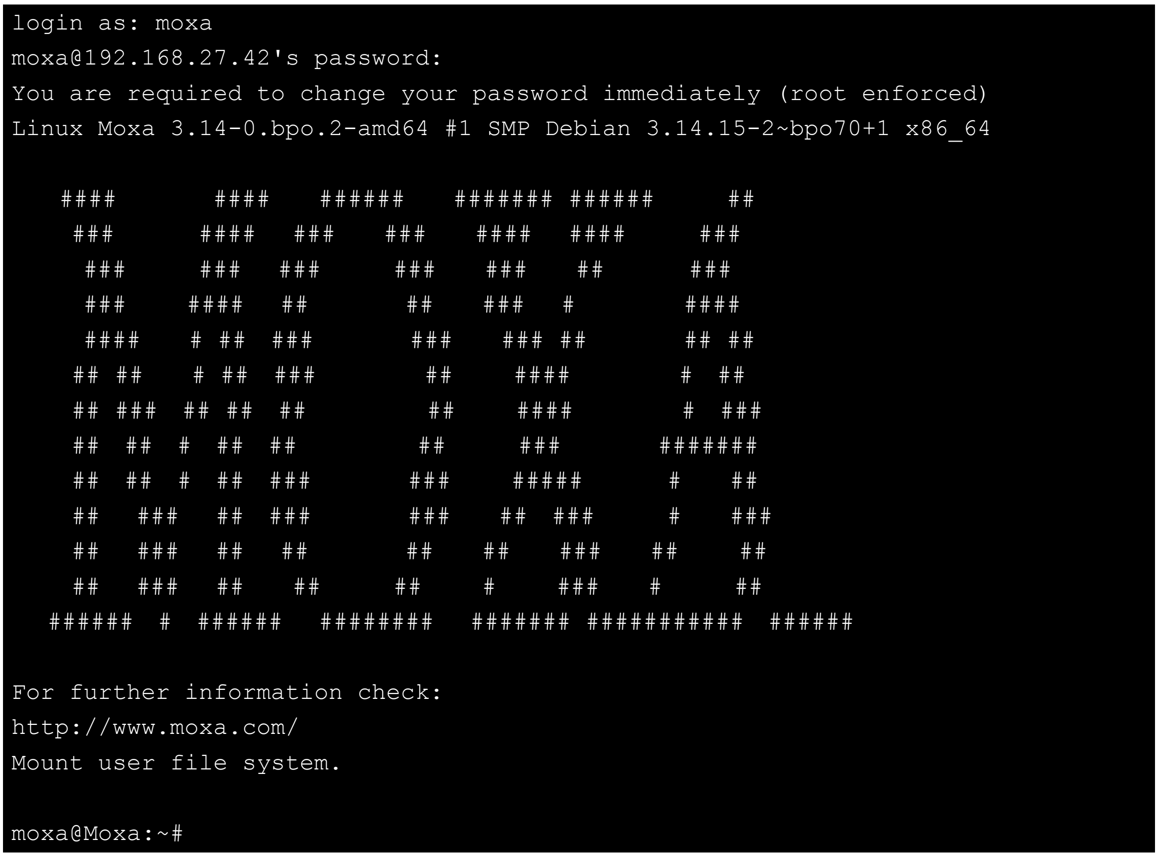
Complete the following steps to log into the DA-820 for the first time.

1. In the login screen, enter the default account information to log in.

   Login: moxa

   Password: moxa

2. When prompted, change the password for the default login account.



```
login as: moxa
moxa@192.168.27.42's password:
You are required to change your password immediately (root enforced)
Linux Moxa 3.14-0.bpo.2-amd64 #1 SMP Debian 3.14.15-2~bpo70+1 x86_64

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

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

moxa@Moxa:~#
```

ATTENTION

For security purposes, it is strongly recommended that you disable the root user and use the sudo command to perform administrative tasks in the DA-920.

Configuring sudo Access

For added security, the **sudo** command provides administrative privileges for trusted users to access the DA-820 without sharing the `root` user password.

Adding a User to the sudo Group

1. Use the **useradd [USER-ACCOUNT-NAME-HERE] sudo** command to add a user to the **sudo** group. For example, **useradd foobar sudo**
2. Use the **group** command to check that the user has been added to the **sudo** group.
3. If required, configure the **sudoers** file.

NOTE For more information about the **sudo** command or the **sudoers** file, refer to the following websites: may

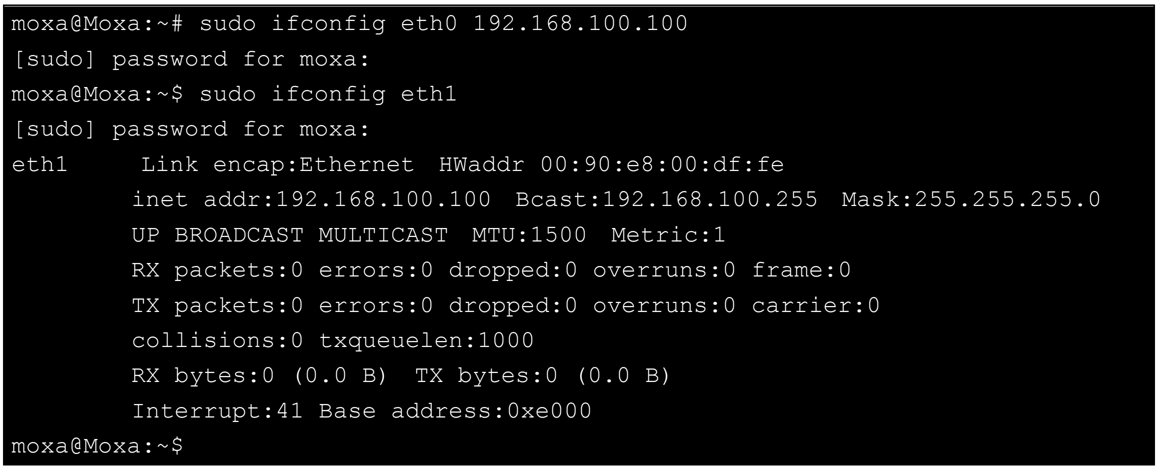
- **Linux.com**'s introduction to **sudo**
  http://www.linux.com/learn/tutorials/306766:linux-101-introduction-to-sudo
- **Debian** introduction to the **sudo** command:
  https://wiki.debian.org/sudo
- **Ubuntu** (a Debian sub-distribution) documentation for the **sudoers** file
  https://help.ubuntu.com/community/Sudoers
- A sample **sudoers** file for an extended network
  http://www.sudo.ws/sudo/sample.sudoers

Using the sudo Command

1. To run commands as the root user, type "sudo" before a command.

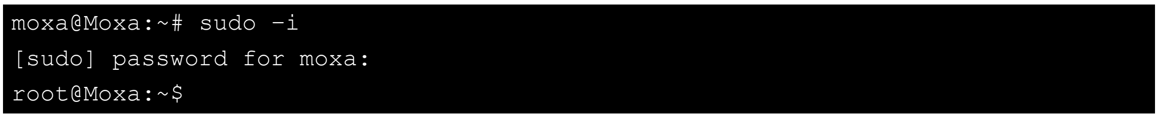
   For example, to configure the IP address on Ethernet port 1, enter the following command:
   sudo ifconfig eth0 192.168.100.100

2. When prompted, enter the password for your user account.



```
moxa@Moxa:~# sudo ifconfig eth0 192.168.100.100
[sudo] password for moxa:
moxa@Moxa:~$ sudo ifconfig eth1
[sudo] password for moxa:
eth1      Link encap:Ethernet  HWaddr 00:90:e8:00:df:fe
         inet addr:192.168.100.100  Bcast:192.168.100.255  Mask:255.255.255.0
         UP BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0  (0.0 B)  TX bytes:0  (0.0 B)
         Interrupt:41 Base address:0xe000
moxa@Moxa:~$
```

To switch to the root user, first disconnect the DA-820 from the network; then, enter the **sudo –I** command.



```
moxa@Moxa:~# sudo -i
[sudo] password for moxa:
root@Moxa:~$
```

Configuring the System and Hardware Clock

This section shows you how to configure the system clock and hardware clock (or Real Time Clock) on the DA-820.

Setting the Time Manually

Use the **`date`** command to display and set the system clock.

`date MMDDhhmmYYYY`

Where:

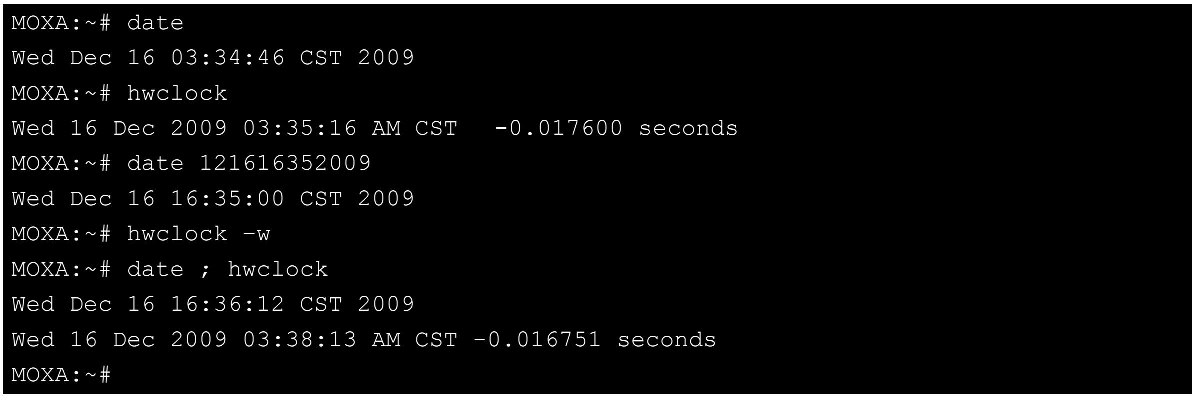
- **`MM`** = Month
- **`DD`** = Date
- **`hh`** = Hour
- **`mm`** = Minute
- **`YYYY`** = Year

To display and set the hardware clock, use the **`hwclock`** command.

The follow command sets the hardware clock to the current system clock.

```
hwclock –w
```

The following figure shows an example.



```
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:~#
```

Synchronizing with a Network Time Server

The DA-820 is pre-installed with an NTP (Network Time Protocol) client. You can use the `ntpdate` command to synchronize the system clock with a network time server.

NOTE Before you set the DA-820 to synchronize with a network time server, make sure that the DA-820 can connect to the Internet.

The following commands synchronize the system clock with the network time server, **`time.stdtime.gov.tw and set the hardware`** clock **`to the system`** clock.

```
root@Moxa:~#ntpdate time.stdtime.gov.tw
root@Moxa:~#hwclock –w
```

For more information on NTP and NTP server, go to http://www.ntp.org.

```
MOXA:~# date ; hwclock
Wed Sep 10 14:51:19 UTC 2014
Wed 10 Sep 2014 02:51:39 PM UTC  -0.938147 seconds
MOXA:~#
MOXA:~# ntpdate time.stdtime.gov.tw
 10 Sep 06:54:14 ntpdate[2510]: step time server 220.130.158.52 offset -28682.196417
sec
MOXA:~#
MOXA:~# hwclock -w
MOXA:~# date ; hwclock
Wed Sep 10 06:54:47 UTC 2014
Wed 10 Sep 2014 06:54:48 AM UTC  -0.734969 seconds
MOXA:~#
```

Using a Script to Update the Time Automatically

You can create and run a shell script to update the time automatically on the DA-820.

The following figure shows a script example.

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

To run the shell script every time the system boots up, complete the following steps:

1. Be default, the root file system is mounted with read-only permission. Re-mount the root file system with read-write permission.

   `root@Moxa:~# mount -o remount,rw /`

2. Save the shell script (for example, **`fixtime`**) in the **`/etc/init.d`** directory and use the **`chmod`** command to set the access mode.

   `root@Moxa:~# chmod 755 fixtime`

3. Open the **`/etc/inittab`** file and append the following line.

   `root@Moxa:~# echo ‘ntp : 2345 : respawn : /etc/init.d/fixtime’ >> /etc/inittab`

4. Use the **`umount /`** command to set the root file directory to read-only mode.

   `root@Moxa:~# umount /`

5. Enter the **`init q`** command to re-initialize the kernel.

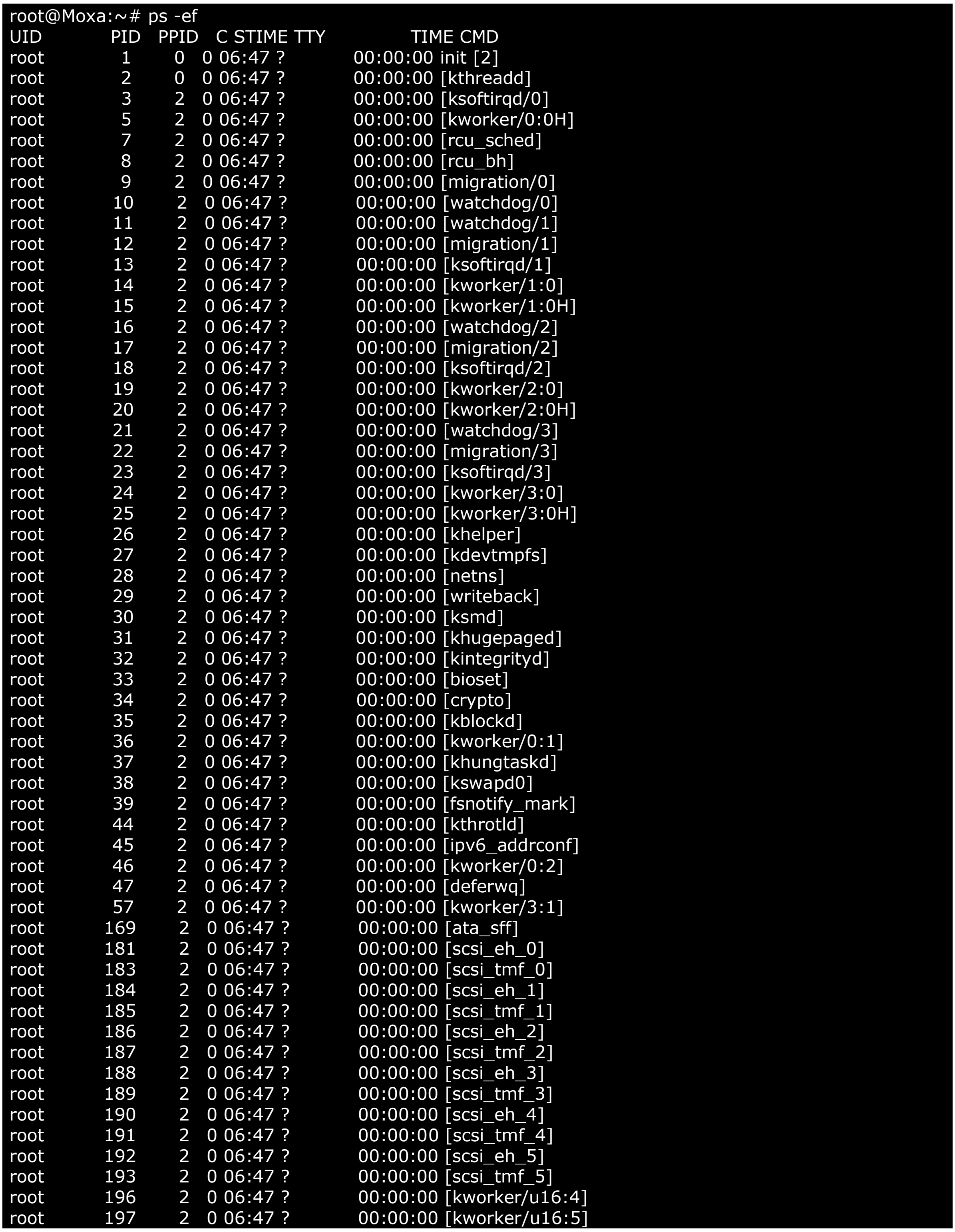
   `root@Moxa:~# init q`

Enabling and Disabling Daemons

When the DA-820 boots up for the first time, the following daemons are enabled by default:

- **ServiceSyncTime:** IRIG-B Time Sync Daemon
- **Inetd:** Internet Daemons
- **Exim4**: SMTP Server Daemon
- **Sshd**: Secure Shell Server Daemon
- **Httpd**: Apache WWW Server Daemon

To display all running processes, enter the **`ps –ef`** command.



```
root@Moxa:~# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 06:47 ?        00:00:00 init [2]
root         2     0  0 06:47 ?        00:00:00 [kthreadd]
root         3     2  0 06:47 ?        00:00:00 [ksoftirqd/0]
root         5     2  0 06:47 ?        00:00:00 [kworker/0:0H]
root         7     2  0 06:47 ?        00:00:00 [rcu_sched]
root         8     2  0 06:47 ?        00:00:00 [rcu_bh]
root         9     2  0 06:47 ?        00:00:00 [migration/0]
root        10     2  0 06:47 ?        00:00:00 [watchdog/0]
root        11     2  0 06:47 ?        00:00:00 [watchdog/1]
root        12     2  0 06:47 ?        00:00:00 [migration/1]
root        13     2  0 06:47 ?        00:00:00 [ksoftirqd/1]
root        14     2  0 06:47 ?        00:00:00 [kworker/1:0]
root        15     2  0 06:47 ?        00:00:00 [kworker/1:0H]
root        16     2  0 06:47 ?        00:00:00 [watchdog/2]
root        17     2  0 06:47 ?        00:00:00 [migration/2]
root        18     2  0 06:47 ?        00:00:00 [ksoftirqd/2]
root        19     2  0 06:47 ?        00:00:00 [kworker/2:0]
root        20     2  0 06:47 ?        00:00:00 [kworker/2:0H]
root        21     2  0 06:47 ?        00:00:00 [watchdog/3]
root        22     2  0 06:47 ?        00:00:00 [migration/3]
root        23     2  0 06:47 ?        00:00:00 [ksoftirqd/3]
root        24     2  0 06:47 ?        00:00:00 [kworker/3:0]
root        25     2  0 06:47 ?        00:00:00 [kworker/3:0H]
root        26     2  0 06:47 ?        00:00:00 [khelper]
root        27     2  0 06:47 ?        00:00:00 [kdevtmpfs]
root        28     2  0 06:47 ?        00:00:00 [netns]
root        29     2  0 06:47 ?        00:00:00 [writeback]
root        30     2  0 06:47 ?        00:00:00 [ksmd]
root        31     2  0 06:47 ?        00:00:00 [khugepaged]
root        32     2  0 06:47 ?        00:00:00 [kintegrityd]
root        33     2  0 06:47 ?        00:00:00 [bioset]
root        34     2  0 06:47 ?        00:00:00 [crypto]
root        35     2  0 06:47 ?        00:00:00 [kblockd]
root        36     2  0 06:47 ?        00:00:00 [kworker/0:1]
root        37     2  0 06:47 ?        00:00:00 [khungtaskd]
root        38     2  0 06:47 ?        00:00:00 [kswapd0]
root        39     2  0 06:47 ?        00:00:00 [fsnotify_mark]
root        44     2  0 06:47 ?        00:00:00 [kthrotld]
root        45     2  0 06:47 ?        00:00:00 [ipv6_addrconf]
root        46     2  0 06:47 ?        00:00:00 [kworker/0:2]
root        47     2  0 06:47 ?        00:00:00 [deferwq]
root        57     2  0 06:47 ?        00:00:00 [kworker/3:1]
root       169     2  0 06:47 ?        00:00:00 [ata_sff]
root       181     2  0 06:47 ?        00:00:00 [scsi_eh_0]
root       183     2  0 06:47 ?        00:00:00 [scsi_tmf_0]
root       184     2  0 06:47 ?        00:00:00 [scsi_eh_1]
root       185     2  0 06:47 ?        00:00:00 [scsi_tmf_1]
root       186     2  0 06:47 ?        00:00:00 [scsi_eh_2]
root       187     2  0 06:47 ?        00:00:00 [scsi_tmf_2]
root       188     2  0 06:47 ?        00:00:00 [scsi_eh_3]
root       189     2  0 06:47 ?        00:00:00 [scsi_tmf_3]
root       190     2  0 06:47 ?        00:00:00 [scsi_eh_4]
root       191     2  0 06:47 ?        00:00:00 [scsi_tmf_4]
root       192     2  0 06:47 ?        00:00:00 [scsi_eh_5]
root       193     2  0 06:47 ?        00:00:00 [scsi_tmf_5]
root       196     2  0 06:47 ?        00:00:00 [kworker/u16:4]
root       197     2  0 06:47 ?        00:00:00 [kworker/u16:5]
root       212      2   0 06:47 ?         00:00:00 [kworker/3:1H]
root       213      2   0 06:47 ?         00:00:00 [kworker/1:1H]
root       214      2   0 06:47 ?         00:00:00 [kworker/1:2]
root       226      2   0 06:47 ?         00:00:00 [md]
root       238      2   0 06:47 ?         00:00:00 [kworker/0:1H]
root       239      2   0 06:47 ?         00:00:00 [jbd2/sda1-8]
root       240      2   0 06:47 ?         00:00:00 [ext4-rsv-conver]
root       264      2   0 06:47 ?         00:00:00 [kworker/2:1H]
root       387      1   0 06:47 ?         00:00:00 udevd --daemon
root       529    387   0 06:47 ?         00:00:00 udevd --daemon
root       557      2   0 06:47 ?         00:00:00 [khubd]
root       580      2   0 06:47 ?         00:00:00 [kpsmoused]
root       595      2   0 06:47 ?         00:00:00 [kvm-irqfd-clean]
root       597      2   0 06:47 ?         00:00:00 [kworker/2:2]
root      2287      1   0 06:47 ?         00:00:00 /usr/sbin/acpid
root      2323      1   0 06:47 ?         00:00:00 /usr/sbin/apache2 -k start
root      2385      1   0 06:47 ?         00:00:00 /sbin/mdadm --monitor --pid-file
root      2462      1   0 06:47 ?         00:00:00 /usr/sbin/cron
root      2610      1   0 06:47 ?         00:00:00 /usr/sbin/pkcsslotd
root      2615      1   0 06:47 ?         00:00:00 /usr/sbin/sshd
www-data  2638   2323   0 06:47 ?         00:00:00 /usr/sbin/apache2 -k start
www-data  2639   2323   0 06:47 ?         00:00:00 /usr/sbin/apache2 -k start
www-data  2640   2323   0 06:47 ?         00:00:00 /usr/sbin/apache2 -k start
www-data  2642   2323   0 06:47 ?         00:00:00 /usr/sbin/apache2 -k start
www-data  2643   2323   0 06:47 ?         00:00:00 /usr/sbin/apache2 -k start
tss       2705      1   0 06:47 ?         00:00:00 /usr/sbin/tcsd
root      2730      1   0 06:47 tty1      00:00:00 /bin/login --
root      2731      1   0 06:47 tty2      00:00:00 /sbin/getty 38400 tty2
root      2732      1   0 06:47 tty3      00:00:00 /sbin/getty 38400 tty3
root      2733      1   0 06:47 tty4      00:00:00 /sbin/getty 38400 tty4
root      2734      1   0 06:47 tty5      00:00:00 /sbin/getty 38400 tty5
root      2735      1   0 06:47 tty6      00:00:00 /sbin/getty 38400 tty6
root      2738    387   0 06:52 ?         00:00:00 udevd --daemon
moxa      2744   2730   0 06:52 tty1      00:00:00 -bash
root      2825   2744   0 06:52 tty1      00:00:00 sudo -i
root      2834   2825   0 06:52 tty1      00:00:00 -bash
root      2880   2615   0 06:53 ?         00:00:00 sshd: moxa [priv]
moxa      2882   2880   0 06:53 ?         00:00:00 sshd: moxa@pts/0
moxa      2883   2882   0 06:53 pts/0     00:00:00 -bash
root      2964   2883   0 06:53 pts/0     00:00:00 sudo -i
root      2973   2964   0 06:53 pts/0     00:00:00 -bash
root      3196   2973   0 06:59 pts/0     00:00:00 ps -ef
```

Setting a Daemon to Run at Startup

You can edit the **rc.local** file to run a daemon at system start-up. Complete the following steps:

1. Be default, the root file system is mounted with read-only permission. Re-mount the root file system with read-write permission.

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

2. Change to the **/etc** directory.

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

3. Type **vi rc.local** to edit the **rc.local** file using the vi editor.

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

4. Add the application daemon that you want to run and save the file.

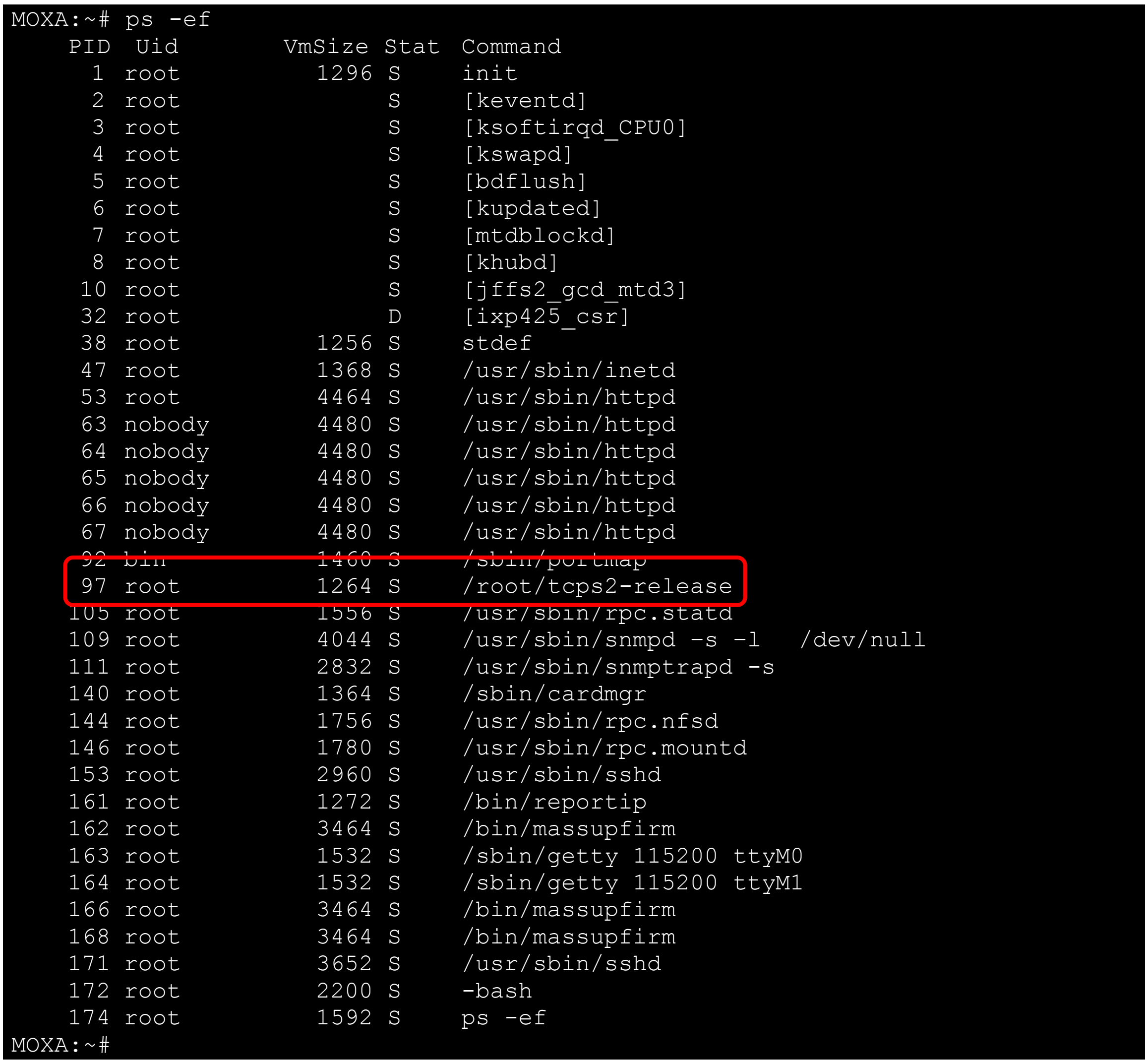
   The follow figure shows an example where the sample program **tcps2-release** (available on the DA-820 software CD/DVD) is added.

```
# !/bin/sh
# Add you want to run daemon
/root/tcp2-release
```

5. Use the **umount /** command to set the root file directory to read-only mode.

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

6. After you restart the DA-820, the application daemon appears on the running process list.



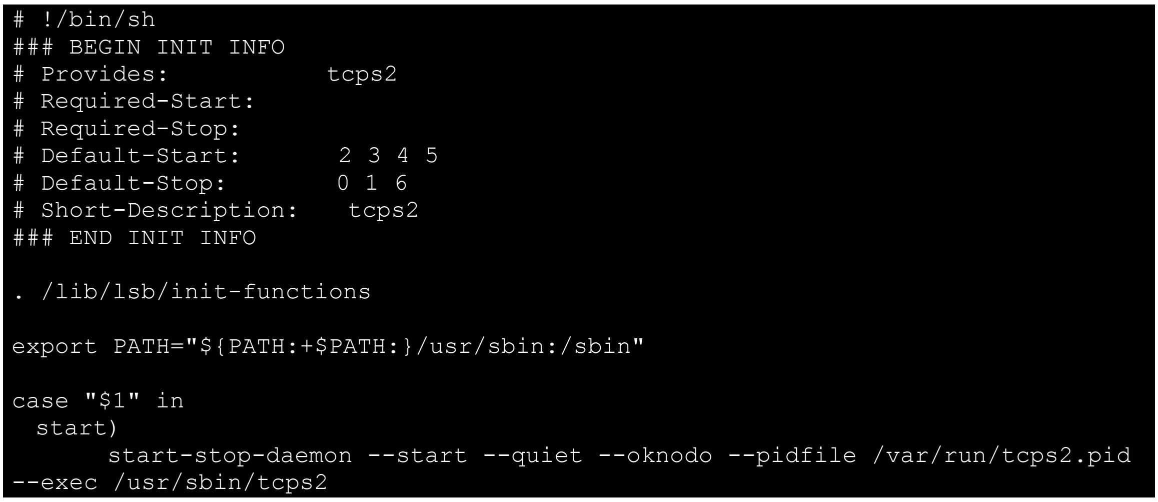
```
MOXA:~# ps -ef
    PID  Uid       VmSize Stat  Command
      1 root         1296 S    init
      2 root              S    [keventd]
      3 root              S    [ksoftirqd_CPU0]
      4 root              S    [kswapd]
      5 root              S    [bdflush]
      6 root              S    [kupdated]
      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 ttyM0
    164 root         1532 S    /sbin/getty 115200 ttyM1
    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:~#
```

Configuring Runlevels Using insserv

You can use the scripts in the **/etc/init.d/** directory to start or stop a service in Linux. To add or remove the service to or from a run level in **/etc/rcX.d/**, use the **insserv** command.

Complete the following steps to set the runlevel of a service.

1. Create a start-stop script as shown in the following figure. Save the script (for example, as **tcps2**) in the **/etc/rcX.d/** directory.



```
# !/bin/sh
### BEGIN INIT INFO
# Provides:           tcps2
# Required-Start:
# Required-Stop:
# Default-Start:       2 3 4 5
# Default-Stop:        0 1 6
# Short-Description:    tcps2
### END INIT INFO

. /lib/lsb/init-functions

export PATH="${PATH:+$PATH:}/usr/sbin:/sbin"

case "$1" in
  start)
       start-stop-daemon --start --quiet --oknodo --pidfile /var/run/tcps2.pid
--exec /usr/sbin/tcps2
```

```
        ;;
  stop)
        start-stop-daemon --stop --quiet --oknodo --pidfile /var/run/tcps2.pid
        ;;
esac

exit 0
```

2. Enter the following `insserv` command to add the service and run the service at startup.

```
moxa@Moxa:~# sudo insserv -v -d tcps2
```

3. Use the ls command to check the runlevel of the service.

```
moxa@Moxa:~# ls -l /etc/rc?.d/*tcps*
lrwxrwxrwx 1 root root 15 Jul  6 09:40 /etc/rc2.d/S18tcps2 -> ../init.d/tcps2
lrwxrwxrwx 1 root root 15 Jul  6 09:40 /etc/rc3.d/S18tcps2 -> ../init.d/tcps2
lrwxrwxrwx 1 root root 15 Jul  6 09:40 /etc/rc4.d/S18tcps2 -> ../init.d/tcps2
lrwxrwxrwx 1 root root 15 Jul  6 09:40 /etc/rc5.d/S18tcps2 -> ../init.d/tcps2
moxa@Moxa:~#
```

To remove a service from all runlevels, enter the **`insser –r`** command.

```
moxa@Moxa:~# insserv -r tcps2
```

Then, use the ls command to check the runlevels of the service. The system should display a warning message.

```
moxa@Moxa:~# ls -l /etc/rc?.d/*tcps*
ls: cannot access /etc/rc?.d/*tcps*: No such file or directory
moxa@Moxa:~#
```

Scheduling Tasks Using cron

You can set the DA-820 to run scheduled tasks using `cron`.

The `cron` daemon searches the **/etc/crontab** directory for crontab files. Every minute, the cron daemon checks each command to see if the command should be run. The output of the commands is sent to the owner of the crontab file (or to the user specified in the MAILTO environment variable in the **crontab** file)

You can schedule command executions in a crontab file. The following table shows the format of a crontab file.

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

For example, the following crontab line sets the system to run a program at 8:00 every day.

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

The following procedure shows the steps to configure cron to update the system time and hardware time every day at 8:00.

1. Create a script with the following lines and save it in the /home/ directory (for example, as fixtime.sh).

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

2. Change the mode of **`fixtime.sh.`**

```
# chmod 755 fixtime.sh
```

3. Append the following line to the **crontab** file to run **`fixtime.sh`** at 8:00 every day.

```
* 8 * * *root    /home/fixtime.sh
```

Mounting a USB Storage Device

The Debian Linux distribution includes a usbmount package that allows you to automatically mount USB storage devices. usbmount depends on the udev demon that calls the script to mount the USB storage devices automatically at specific mount points. The first USB storage device will be mounted on **/media/usb0**, and the second USB storage device on **/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:~#
```

Note that usbmount is only available in text mode and does not support the gnome desktop environment. Alternatively, you can install gnome-volume-manager for better support.

```
MOXA:~# mount -o,remount rw /
MOXA:~# apt-get remove usbmount
MOXA:~# apt-get install gnome-volume-manager
MOXA:~# umount /
```

ATTENTION

To prevent data loss, before you disconnect a USB storage device from the DA-820, enter the sync command.

ATTENTION

Before you disconnect a USB storage device, exit from the mount directory (for example, /media/usb0 or /media/usb1); otherwise, the automatic un-mount process will not function.

If the automatic un-mount process fails to function, use the umount command to manually unmount the USB storage devices.

Checking the Linux Version

You can use the **uname** (Unix Name) command to view information (for example, name and version) about the Linux distribution installed on the DA-820.

The following figure shows an example.

```
MOXA:~# uname -a
Linux Moxa 3.14-0.bpo.2-amd64 #1 SMP Tue Jun 14 09:42:28 UTC 2011 x86_64 GNU/Linux
MOXA:~#
```

Installing and Removing Packages Using APT

In Debian, you can use APT (Advanced Package Tool) to install and remove packages.

Installing a Package

Complete the following steps to install a package using APT.

1. Before you install a package using APT, you must configure the apt source list **/etc/apt/sources.list** that is read-only be default.

   Mount the root file system with write permission.

```
MOXA:~# mount -o remount,rw
```

2. Edit the **/etc/apt/sources.list** file in the vi editor.

```
MOXA:~# vi /etc/apt/sources.list
#
# deb cdrom:[Debian GNU/Linux 7.6.0 _Wheezy_ - Official amd64 NETINST Binary-1
20130615-23:04]/ wheezy main
deb http://ftp.us.debian.org/debian/ wheezy main
deb-src http://ftp.us.debian.org/debian/ wheezy main
deb http://security.debian.org/ wheezy/updates main
deb-src http://security.debian.org/ wheezy/updates main
# wheezy-updates, previously known as ‘volatile’
deb http://ftp.us.debian.org/debian wheezy-updates main
deb-src http://ftp.us.debian.org/debian wheezy-updates main
```

3. Enter the **apt-get** update command to update the APT source list.

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

4. Enter the **apt-get** command with the **install** option to install a package (for example, openswan).

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

5. After the installation is complete, 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 want to install a huge package, link /var/cache/apt to the USB drive or mount it to an NFS space to generate more free space. Use the df –h command to check how much free space is available on tmpfs.

```
Moxa:~# df -h
Filesystem          Size  Used Avail Use% Mounted on
rootfs             1.5G 1001M  440M  70% /
udev                10M  748K  9.3M   8% /dev
/dev/sda1           1.5G 1001M  440M  70% /
tmpfs              501M     0  501M   0% /lib/init/rw
tmpfs              501M     0  501M   0% /dev/shm
none               501M   19M  482M   4% /tmp
/dev/sda2           270M  130M  126M  51% /home
Moxa:~#
```

Removing a Package

You can use one of the following APT commands to remove a package:

- To remove a package without deleting related configuration files.

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

- To remove a package and delete all related configuration files.

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

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

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

Setting up a Desktop Environment

This section shows you how to set up a desktop environment in Debian on the DA-820. By default, the Debian Linux distribution on the DA-820 is not pre-installed with a desktop environment. Debian supports various full-featured graphical environments (such as Gnome and KDE).

To install a desktop environment on the DA-820, use one of the following commands:

- Gnome

```
moxa@MOXA:~# sudo apt-get install gnome-desktop
```

- KDE

```
moxa@MOXA:~# sudo apt-get install kde-standard
```

- Xfce

```
moxa@MOXA:~# apt-get install xfce4 xfce4-goodies thunar-archive-plugin
```

- LXDE

```
moxa@MOXA:~# sudo apt-get install lxde-core lxde
```

3

Managing Communications

The DA-820-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:

Renaming a Network Interface

Linux systems use **udevd** to detect new network interfaces (for example, Ethernet and wireless interfaces). However, because the automatically assigned name for an interface may not match the label, you can specify the naming rules in **/etc/udev/rules.d/01-rename_net_interface.rules**.

The following figure shows an example of the rule content.

```
KERNEL=="eth*", KERNELS=="0000:00:19.0", NAME="eth3"
KERNEL=="eth*", KERNELS=="0000:01:00.0", NAME="eth0"
KERNEL=="eth*", KERNELS=="0000:03:00.0", NAME="eth2"
KERNEL=="eth*", KERNELS=="0000:02:00.0", NAME="eth1"
```

Configuring Network Settings

There are four 10/100/1000 Ethernet ports on the DA-820-LX computer. The following table shows the default IP addresses and subnet masks on the Ethernet ports.

	Default IP address	Subnet mask
LAN1	192.168.3.127	255.255.255.0
LAN2	192.168.4.127	255.255.255.0
LAN3	192.168.5.127	255.255.255.0
LAN4	192.168.6.127	255.255.255.0

You can edit the interface configuration file interface to permanently change the network settings.

To change network settings temporarily, use the **ipconfig** command.

Editing an Interface Configuration File

1. Type **cd /etc/network** to change directories.

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

2. Type **vi interfaces** to edit the file in the vi editor. Refer to the following sections for information on how to set an Ethernet port to use a static or dynamic (DHCP) IP address.

```
MOXA:/etc/network# vi interfaces
```

3. Type the following command to make the network settings take effect.

   # /etc/init.d/networking restart

Using Static IP Addresses

The following figure shows a configuration file example where the Ethernet ports are set to use static IP address that you can change.

```
# 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
auto eth2
iface eth2 inet static
       address 192.168.5.127
       netmask 255.255.255.0
       broadcast 192.168.5.255

auto eth3
iface eth3 inet static
       address 192.168.6.127
       netmask 255.255.255.0
       broadcast 192.168.6.255
```

Using Dynamic IP Addresses with DHCP

The following figure shows an example where the **eth0** interface is set to request for a dynamic IP address from a DHCP server.

```
# The primary network interface
auto eth0
iface eth0 inet dhcp
```

After you change the configuration file, enter the following command to make the changes take effect immediately.

# /etc/init.d/networking restart

```
MOXA:~# /etc/init.d/networking restart
```

Adjusting IP Addresses Using ifconfig

You can temporarily change the IP address of an Ethernet port using the **ipconfig** command. IP address settings made using the **ipconfig** command is not saved on the system ROM.

The following figure shows an example where the Ip address of the **eth0** port is change to 192.168.1.1.

```
MOXA:~# ifconfig eth0 192.168.1.1
MOXA:~#
```

Configuring Telnet and TFTP Servers

In addition to supporting Telnet client/server and TFTP client/server, the DA-820-LX also supports SSH and sftp client/server.

To enable or disable the Telnet or ftp server, you must edit the file **/etc/inetd.conf**.

1. Mount the root file system with write permission.

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

2. Type **cd /etc** to change the directory.

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

3. Type **vi inetd.conf** to edit the configuration file in the vi editor.

   ```
   MOXA:/etc# vi inetd.conf
   ```

Enabling the Telnet or TFTP Server

The following example shows the content of the file **/etc/inetd.conf**. By default, the Telnet and TFTP servers are enabled.

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

Disabling the Telnet or TFTP Server

To disable the Telnet or TFTP server, add an "**#**" prefix to an entry in the configuration file.

The following figure shows an example where the TFTP server is disabled.

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

After you finish editing the interface configuration file, unmount the root directory back to read-only mode and restart the **inetd** service.

```
MOXA:~# umount /
MOXA:~# service openbsd-inetd restart
```

Setting up DNS Client

To configure the DNS client on the DA-820, edit the following configuration files:

- **/etc/hostname**
- **/etc/resolv.conf**
- **/etc/nsswitch.conf**

/etc/hostname

1. Edit the /etc/hostname file in vi editor.

```
moxa@MOXA:~# sudo vi /etc/hostname
MOXA
```

2. Re-configure the hostname.

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

3. Check the new hostname.

```
moxa@MOXA:~# hostname
```

/etc/resolv.conf

The **resolv.conf** file contains the information that allows the system to resolve names into IP addresses. For example, before you can set the system time to synchronize with the **time.stdtime.gov.tw** NTP server, you must add a DNS server IP address in the **resolv.conf** file.

Check with your network administrator to obtain the DNS server IP address information.

You must add an entry in the **resolv.conf** file in the format: **nameserver** [*IP address*]

The following figure shows an example of the **resolv.conf** file content.

```
MOXA:/etc# cat resolv.conf
#
# resolv.conf  This file is the resolver configuration file
# See resolver(5).
#
#nameserver 192.168.1.16
nameserver 168.95.1.1
nameserver 140.115.1.31
nameserver 140.115.236.10
MOXA:/etc#
```

/etc/nsswitch.conf

The **nsswitch.conf** file defines the reading sequence of the **/etc/hosts** or **/etc/resolv.conf** file to resolve the IP address.

In the following example, the sequence setting for **hosts** sets the system to check the **/etc/host** file first and then use the DNS service to resolve IP addresses.

```
# /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
```

Configuring Ethernet Bonding

You can use the bonding driver in Linux to aggregate multiple network interfaces into a single logical "bonded" interface. To use the bonding feature, first load the bonding driver with mode settings. Then, use the **ifenslave** command to add two or more Ethernet interfaces to the **bond0** interface.

The following figure shows a sample script (**/etc/init.d/bonding.sh**) that aggregates interfaces **eth1** and **eth2** into interface **bond0**.

```
#! /bin/bash

#### BEGIN INIT INFO
# Provides:          bonding
# Short-Description: Start the bonding service, bond eth1 and eth2.
# Required-Start:    $all
# Required-Stop:     $all
# Should-Start:
```

```
# Should-Stop:
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
### END INIT INFO

NAME=bonding
PATH=/bin:/usr/bin:/sbin:/usr/sbin

case "$1" in
  start)
    # to set ethX interfaces as slave the bond0 must have an ip
    if [ "$2" == "" ]; then
      $0
      exit 1
    fi
    echo "Starting bonding service: $NAME."
    modprobe bonding mode=1 miimon=100      # load bonding module

    ifdown eth2                  # putting down eth2
    ifdown eth1                  # putting down eth1

    ifconfig bond0 hw ether 00:90:E8:00:00:60   # change mac address
    ifconfig bond0 $2 netmask 255.255.255.0 up  # set ip address

    ifenslave bond0 eth2         # set eth2 in slave for bond0
    ifenslave bond0 eth1         # set eth1 in slave for bond0
    ;;

  stop)
    echo "Stopping bonding service: $NAME"
    ifenslave -d bond0 eth2          # release eth2 from bond0
    ifenslave -d bond0 eth1          # release eth1 from bond0

    ifconfig bond0 down              # putting down bond0
    modprobe -r bonding              # unload bonding module

    ifup eth2
    ifup eth1
    ;;

  restart)
    $0 stop
    $0 start $2
    ;;

  *)
    echo "Usage: /etc/init.d/$NAME {start|stop|restart} [ip address]"
    exit 1
    ;;
esac

exit 0
```

Use the **insserv** command to set the runlevel of the **bonding.sh** script.

```
moxa@MOXA:~# sudo insserv -v -d bonding.sh
```

Use the **insserv –r** command to remove the script from all runlevels.

```
moxa@MOXA:~# sudo insserv -r bonding.sh
```

Configuring the Apache Web Server

Default Homepage

The configuration file for the Apache web server is **/etc/apache2/sites-enabled/000-default** and the default homepage file is **/var/www/apache2-default/index.html**.

You can save the files for your homepage in the following directory:

/var/www

Save your CGI pages in the following directory:

/var/www

Before you modify the homepage, complete the following steps to test whether the Apache web server on the DA-820 is running:

1. Open a web browser (for example, Microsoft Internet Explorer or Mozilla Firefox). Then, enter the LAN IP address of the DA-820 in the address bar (for example, http://192.168.3.127)
2. Test the default CGI page. Enter the address in the format:
   http://[*IP address*]/cgi-bin/w3mmail.cgi

   For example, http://192.168.3.127/cgi-bin/w3mmail.cgi.

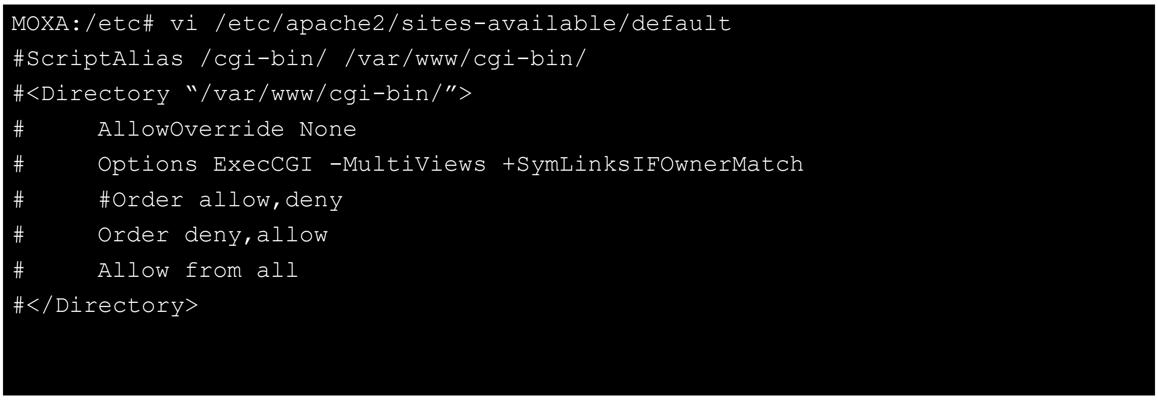
Disabling the CGI Function

By default, the CGI function is enabled. If you want to disable this function, edit the **/etc/apache2/sites-enabled/000-default** file.

1. Type **vi/etc/apache2/sites-enabled/000-default** to open the configuration file in vi.
2. Type "#" to comment out the following lines:

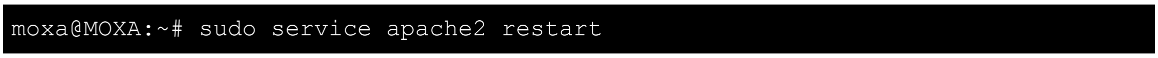
```
#ScriptAlias /cgi-bin/ /var/www/cgi-bin/
#<Directory "/var/www/cgi-bin/">
# AllowOverride None
# Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
# #Order allow,deny
# Order deny,allow
# Allow from all
#</Directory>
```

   The following figure shows an example.



```
MOXA:/etc# vi /etc/apache2/sites-available/default
#ScriptAlias /cgi-bin/ /var/www/cgi-bin/
#<Directory “/var/www/cgi-bin/”>
#     AllowOverride None
#     Options ExecCGI -MultiViews +SymLinksIFOwnerMatch
#     #Order allow,deny
#     Order deny,allow
#     Allow from all
#</Directory>
```

3. Re-start the Apache web server.



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

ATTENTION

If you develop your own CGI applications, make sure that the CGI file is executable.

Saving Web Pages to a USB Storage Device

If you save web pages on a USB storage device, you can configure the Apache web server to open web pages saved on the USB storage device. You can obtain samples files shown in this section from the Moxa website.

1. Prepare the web pages and then save the pages on a USB storage device.

   You can download the web page test suite from the web site at **http://www.w3.org/MarkUp/Test/HTML401.zip**.

1. Uncompress the zip file to your computer and transfer the files (for example, using FTP) to the **/media/usb0** directory on the DA-820.
2. Mount the root file system with write permission.

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

3. Type **vi /etc/apache2/sites-avaliable/default** and **vi /etc/apache2/sites-avaliable/default-ssl** to edit the configuration files.

```
MOXA:/etc# vi /etc/apache2/sites-avaliable/default
MOXA:/etc# vi /etc/apache2/sites-avaliable/default-s
```

4. Change the **DocumentRoot** directory to the USB storage directory (for example, /media/usb0/www).

```
<VirtualHost *:80>
...
        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>
“etc/apache2/sites-avaliable/default”
<VirtualHost *:443>
...
        DocumentRoot /media/usb0/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
...
        ScriptAlias /cgi-bin/ /media/usb0/www/cgi-bin/
        <Directory “/media/usb0/wwwz/cgi-bin/”>
                AllowOverride None
                Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>
...
</VirtualHost>
/etc/apache2/sites-avaliable/default-ssl”
```

5. Use the following commands to restart the Apache web server:

   cd /etc/init.d
   ./apache2 restart

6. Start a web browser on your computer and type the LAN IP address of the DA-820-LX in the address bar.
7. Unmount the root directory back to read-only mode.

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

8. Re-start the Apache web server.

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

NOTE For more information on configuring the Apache web server, go to the Apache website at http://httpd.apache.org/docs/.

Configuring IPTABLES

You can use the IPTABLES package to configure packet filtering, network address translation (NAT), firewall, and packet mangling in Linux.

The DA-820-LX supports three types of IPTABLES: Filter tables, NAT tables, and Mangle tables. Each table contains built-in chains and user-defined chains. Each chain is a list of rules that apply to a certain type of packet. 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) is called a **target**.

Filter Table

The Filter table contains the following chains:

- **INPUT chain**
- **OUTPUT chain**
- **FORWARD chain**

NAT Table

The NAT table contains the following chains:

- **PREROUTING chain**—Translates the destination IP address (DNAT).
- **POSTROUTING chain**—Works after the routing process and before the Ethernet device process to translate the source IP address (SNAT).
- **OUTPUT chain**—Creates local packets.

Sub-tables

The Sub table contains the following chains:

- **Source NAT (SNAT)**—Changes the first source IP address of the packet.
- **Destination NAT (DNAT)**—Changes the first destination IP address of the packet.
- **MASQUERADE**—A special form for SNAT. If one host can connect to the Internet, then the other computers that connect to this host can connect to the Internet when the computer does not have an actual IP address.
- **REDIRECT**—A special form of DNAT that re-sends packets to a local host independent of the destination IP address.

Mangle Table

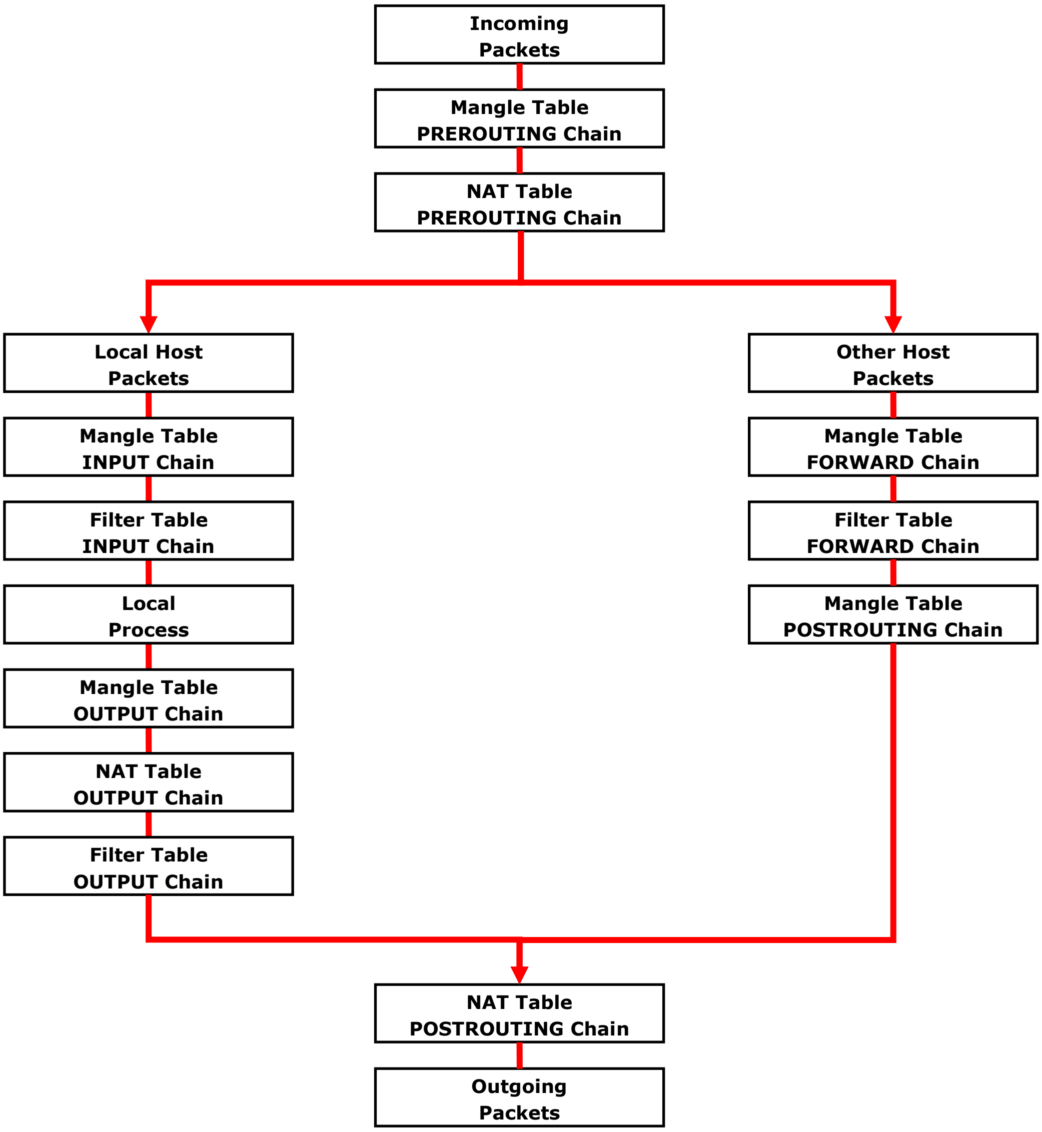
The Mangle table contains the following chains:

- **PREROUTING chain**—Pre-processes packets before the routing process.
- **OUTPUT chain**—Processes packets after the routing process.

Mangle tables can have one of three extensions—TTL, MARK, TOS.

IPTABLES Hierarchy

The following figure shows the IPTABLES hierarchy.


Incoming Packets
Mangle Table PREROUTING Chain
NAT Table PREROUTING Chain
Local Host Packets
Mangle Table INPUT Chain
Filter Table INPUT Chain
Local Process
Mangle Table OUTPUT Chain
NAT Table OUTPUT Chain
Filter Table OUTPUT Chain
Other Host Packets
Mangle Table FORWARD Chain
Filter Table FORWARD Chain
Mangle Table POSTROUTING Chain
NAT Table POSTROUTING Chain
Outgoing Packets

IPTABLES Modules

The DA-820-LX supports the following sub-modules. Make sure that you use the module that matches your application.

arptable_filter.ko	arp_tables.ko	arpt_mangle.ko	ip_conntrack_amanda.ko
ip_conntrack_ftp.ko	ip_conntrack_h323.ko	ip_conntrack_irc.ko	ip_conntrack.ko
ip_conntrack_netbios_ns.ko	ip_conntrack_netlink.ko	ip_conntrack_pptp.ko	ip_conntrack_proto_sctp.ko
ip_conntrack_sip.ko	ip_conntrack_tftp.ko	ip_nat_amanda.ko	ip_nat_ftp.ko
ip_nat_h323.ko	ip_nat_irc.ko	ip_nat.ko	ip_nat_pptp.ko
ip_nat_sip.ko	ip_nat_snmp_basic.ko	ip_nat_tftp.ko	ip_queue.ko
iptable_filter.ko	iptable_mangle.ko	iptable_nat.ko	iptable_raw.ko
ip_tables.ko	ipt_addrtype.ko	ipt_ah.ko	ipt_CLUSTERIP.ko
ipt_dscp.ko	ipt_DSCP.ko	ipt_ecn.ko	ipt_ECN.ko
ipt_hashlimit.ko	ipt_iprange.ko	ipt_LOG.ko	ipt_MASQUERADE.ko
ipt_NETMAP.ko	ipt_owner.ko	ipt_recent.ko	ipt_REDIRECT.ko
ipt_REJECT.ko	ipt_SAME.ko	ipt_TCPMSS.ko	ipt_tos.ko
ipt_TOS.ko	ipt_ttl.ko	ipt_TTL.ko	ipt_ULOG.ko

The basic syntax to enable and load an IPTABLES module is as follows:

```
# lsmod
# modprobe ip_tables
# modprobe iptable_filter
# modprobe iptable_mangle
# modprobe iptable_nat
```

Use the **lsmod** command to check whether the **ip_tables** module has already been loaded in the DA-820-LX. Use the **modprobe** command to insert and enable the module.

Use the **iptables**, **iptables-restore**, and **iptables-save** commands to maintain the database.

ATTENTION

IPTABLES rules are used for packet filtering and NAT. Make sure that you set the IPTABLES rules correctly. An improper rule may prevent a remote host from connecting to the DA-820 on the LAN or through PPP.
It is recommended that you use the VGA console to configure IPTABLES.
For more information on IPTABLES, go to the following web sites:

http://www.linuxguruz.com/iptables/
http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html

The following sections show you how to configure and manage IPTABLES rules.

Viewing Rules and Deleting Chains

This section describes the iptables commands to view rules and delete chains.

Usage

iptables [-t tables] [-L] [-n]

-t tables: Table to manipulate (default: 'filter'). For example, nat or filter.

-L [chain]: List all rules in selected chains. If no chain is selected, all chains are listed.

-n: Numeric output of addresses and ports.

iptables [-t tables] [-FXZ]

-F: Flush the selected chain (all the chains in the table if none is listed).

-X: Delete the specified user-defined chain.

-Z: Set the packet and byte counters in all chains to zero.

Example

iptables -L -n

In this example, since we do not use the -t parameter, the system uses the default "filter" table. Three chains are included: INPUT, OUTPUT, and FORWARD. INPUT chains are accepted automatically, and all connections are accepted without being filtered.

iptables –F
iptables –X
iptables -Z

Defining Chain Policies

This section describes the commands you use to define chain policies.

Usage

iptables [-t tables] [-P] [INPUT, OUTPUT, FORWARD, PREROUTING, OUTPUT, POSTROUTING] [ACCEPT, DROP]

-P: Set the policy for the chain to the given target.

INPUT: For packets coming into the DA-820-LX.

OUTPUT: For locally-generated packets.

FORWARD: For packets routed out through the DA-820-LX.

PREROUTING: To alter packets as soon as they come in.

POSTROUTING: To alter packets as they are about to be sent out.

Example

#iptables –P INPUT DROP
#iptables –P OUTPUT ACCEPT
#iptables –P FORWARD ACCEPT
#iptables –t nat –P PREROUTING ACCEPT
#iptables –t nat –P OUTPUT ACCEPT
#iptables -t nat –P POSTROUTING ACCEPT

In this example, the policy accepts outgoing packets and denies incoming packets.

Adding or Deleting Rules

This section describes the commands you use to add or delete rules.

Usage

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: Name of an interface via which a packet is going to be received.

-o: Name of an interface via which a packet is going to be sent.

-p: The protocol of the rule or of the packet to check.

-s: Source address (network name, host name, network IP address, or plain IP address).

--sport: Source port number.

-d: Destination address.

--dport: Destination port number.

-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.

Examples

Example 1: Accept all packets from the lo interface.

```
# iptables –A INPUT –i lo –j ACCEPT
```

Example 2: Accept TCP packets from 192.168.0.1.

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.0.1 –j ACCEPT
```

Example 3: Accept TCP packets from Class C network, 192.168.1.0/24.

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.1.0/24 –j ACCEPT
```

Example 4: Drop TCP packets from 192.168.1.25.

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.1.25 –j DROP
```

Example 5: Drop TCP packets addressed for port 21.

```
# iptables –A INPUT –i eth0 –p tcp --dport 21 –j DROP
```

Example 6: Accept TCP packets from 192.168.0.24 to DA-820-LX’s port 137, 138, 139

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.0.24 --dport 137:139 –j ACCEPT
```

Example 7: Log TCP packets that visit DA-820-LX’s port 25.

```
# iptables –A INPUT –i eth0 –p tcp --dport 25 –j LOG
```

Example 8: Load the ipt_mac module and drop all packets from MAC address 01:02:03:04:05:06.

```
# modprobe ipt_mac

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

ATTENTION

For Example 8, you must first use the command **# modprobe ipt_mac** to load the **ipt_mac** module.

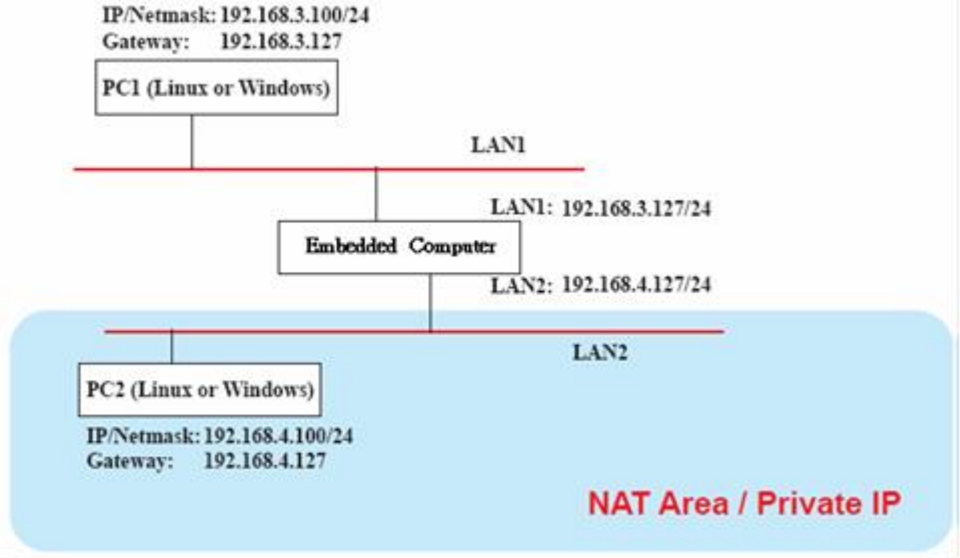
NAT (Network Address Translation)

The NAT (Network Address Translation) protocol translates IP addresses used on one 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 DA-820-LX connects several devices on a network and maps local inside network addresses to one or more global outside IP addresses, and un-maps the global IP addresses on incoming packets back into local IP addresses.

NOTE For more information on NAT, go to the web site at http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html.

NAT Example

The following figure shows a network example where the IP address of all packets leaving **LAN1** are changed to **192.168.3.127** (using the **ipt_MASQUERADE** module).



Enabling NAT during System Startup

You can create a shell script to enable NAT during startup. The following shows an example script.

```
#!/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
```

PPP (Point to Point Protocol)

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 or PPP access is almost identical to connecting directly to a network through the DA-820-LX Ethernet port. Since PPP is a peer-to-peer system, the DA-820-LX can also use PPP to link two networks (or a local network to the Internet) to create a Wide Area Network (WAN).

NOTE For more information on PPP, go to the following web sites:

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

Connecting to a PPP Server Using a Dial-up Connection

You can use the following command to connect to a PPP server through a 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 192.1.1.17" are optional.

```
#pppd connect 'chat -v "" ATDT5551212 CONNECT "" ' ogin: username word: password' /dev/
ttyS0 115200 debug crtscts modem defaultroute 192.1.1.17
```

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

```
#pppd connect 'chat -v "" ATDT5551212 CONNECT "" ' user username password password /dev/
ttyS0 115200 crtscts modem
```

The following describes the parameters and options for the pppd command.

connect 'chat etc...' This option sets the system to contact the PPP server.

The **chat** program is used to dial into 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).

ogin: 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.

You must use this option if the baudrate is set at 115200.

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

defaultroute After the PPP link is established, set it as the default route (especially if the PPP link connects to the Internet).

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 Hard-wired Link

If you have to enter a username and password, use the following command (note that **noipdefault** is optional):

```
pppd connect ‘chat –v” “ “ “ ‘ noipdefault /dev/ttyS0 19200 crtscts
```

If you do not have to enter a username and password, use the following command (note that **noipdefault** is optional, and the username and password are both “root”):

```
pppd connect ‘chat –v” “ “ “ ‘ user root password root noipdefault /dev/ttyS0 19200
crtscts
```

Checking the Connection

After you have set up a PPP connection, you can use various methods to test the connection.

First, use the ipconfig command.

```
# /sbin/ifconfig
```

Depending on your distribution, the command might be located in a different directory. After executing the command, you should be able to see all of the network interfaces that are up.

The following figure shows an output example.

ppp0 should be one of the network interfaces. **inet addr** indicates the IP address of the computer, and **P-t-P** indicates the IP address of the server.

```
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
```

Next, test using the ping command.

```
# ping z.z.z.z
```

where z.z.z.z is the address of a server. The output should be similar to the following:

```
MOXA:~# 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:~#
```

Then, use the **netstat** command to display the routing table.

```
# 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 to test the connection. Note that you will have to use numeric IP addresses unless you have configured **/etc/resolv.conf** correctly.

Configuring the Server for Incoming PPP Connections

Method 1: pppd dial-in with pppd commands

Use this method if you are using a modem and require login authentication.

```
#pppd /dev/ttyS0 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 (*) allows everyone to log in. The second star (*) allows every host to connect. The pair of double quotation marks (“”) indicates that the file **/etc/passwd** can be used to check the password. The last star (*) allows any remote IP addresses to connect.

The following example does not check the username and password:

```
# pppd/dev/ttyS0 115200 crtscts modem 192.168.16.1:192.168.16.2
```

Method 2: pppd dial-in with pppd script

Create a dial-in script **/etc/ppp/peer/dialin** as shown in the following figure.

```
# 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-ttyS0.chat”

# Set up routing to go through this PPP link
defaultroute

# Default modem (you better replace this with /dev/ttySx!)
/dev/ttyS0

# 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-ttyS0.chat** as shown in the following figure.

```
SAY     ‘Auto Answer ON\n’
``      ATS0=1
```

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

```
# pppd call dialin
```

NOTE If you want to set up auto dial-in service, use the respawn command to execute the dial-in service in /etc/inittab. The following figure shows an example.

```
MOXA:~# mount –o remount,rw /dev/sda1 /
MOXA:~# echo “p0:2345:respawn:pppd call dialin” >> /etc/inittab
MOXA:~# umount /
```

Configuring PPPoE

Complete the following steps to configure PPPoE:

1. Connect the DA-820-LX's LAN port to an ADSL modem directly using a cross-over cable, or through a HUB or switch.
2. Log in to the DA-820-LX as the root user.
3. Open the **/etc/ppp/chap-secrets** file in vi and add the following line:
   "username@hinet.net"   *   "password" *

   In the following example, **username@hinet.net** is the username obtained from the ISP and **password** is the corresponding password for the account.

```
# Secrets for authentication using CHAP
# client       server  secret               IP addresses

# PPPOE example, if you want to use it, you need to unmark it and modify it
"username@hinet.net"   *      "password"     *
```

4. Edit the file **/etc/ppp/pap-secrets** and add the following line:
   "username@hinet.net"   *   "password" *

```
# ATTENTION: The definitions here can allow users to log in 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@hinet.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@hinet.net is the username obtained from the ISP to log in to the ISP account. **password** is the corresponding password for the account.

5. Edit the file **/etc/ppp/options** and add the following line:
   plugin rp-pppoe

```
# received.  Note: it is not advisable to use this option with the persist
# option without the demand option.  If the active-filter option is given,
# data packets which are rejected by the specified activity filter also
# count as the link being idle.
#idle <n>

# Specifies how many seconds to wait before re-initiating the link after
# it terminates.  This option only has any effect if the persist or demand
# option is used.  The holdoff period is not applied if the link was
# terminated because it was idle.
#holdoff <n>

# 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>---
```

6. If you use LAN1 to connect to the ADSL modem, add the file **/etc/ppp/options.eth0**. Similarly, if you use LAN2, add **/etc/ppp/options.eth1**.

```
name username@hinet.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.

7. Set up DNS.

If you are using DNS servers supplied by your ISP, add the nameserver information in the file **/etc/resolv.conf** in the following format:

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

Use the following command to create a **pppoe** connection:
pppd eth0

8. The ADSL modem is connected to the **LAN1** port, use **eth0**. If the ADSL modem is connected to **LAN2**, use **eth1**.And similarly for other ports.
9. Type **ifconfig ppp0** to check if the connection is OK. If the connection is OK, you should see the IP address of ppp0. Use **ping** 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
```

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

NFS (Network File System) Client

You can use Network File System (NFS) to mount a disk partition on a remote machine (as if it were on a local hard drive). This allows fast, seamless sharing of files across a network. NFS enables you to develop applications for the DA-820-LX without worrying about the amount of disk space that will be available. The DA-820-LX only supports NFS client protocol.

NOTE For more information on NFS, visit the following web sites:

http://www.ietf.org/rfc/rfc1213.txt
http://www.faqs.org/rfcs/rfc1317.html

Complete the following steps to mount a remote NFS server:

1. Scan the NFS Server's shared directory:

   ```
   #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

2. Establish a mount point on the NFS Client site.

   ```
   #mkdir  –p  /home/nfs/public
   ```

3. Mount the remote directory to a local directory.

   ```
   # mount -t nfs -o nolock 192.168.3.100:/home/public /home/nfs/public
   ```

   (where 192.168.3.100 is the IP address of the NFS server.)

OpenVPN

OpenVPN provides two types of tunnels for users to implement VPNS: **Routed IP Tunnels** and **Bridged Ethernet Tunnels**.

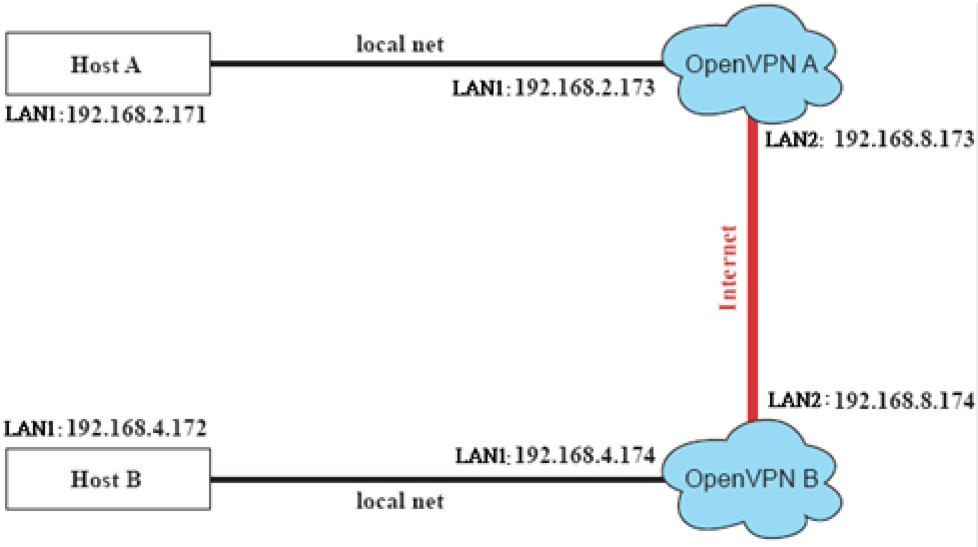
An Ethernet bridge is used to connect different Ethernet networks together. The Ethernets are bundled into one bigger, "logical" Ethernet. Each Ethernet corresponds to one physical interface (or port) that is connected to the bridge.

On each OpenVPN machine, set the configuration files in the **/etc/openvpn** directory, where script files and key files are stored. After an OpenVPN connection is established, the system performs all network data operations in the **/etc/openvpn** directory.

Ethernet Bridging for Private Networks on Different Subnets

This section describes the steps to configure Ethernet bridging for the example network shown in the following figure.

In this example, Host A represents the machine that belongs to OpenVPN A, and Host B belongs to OpenVPN B. The two remote subnets are configured for a different range of IP addresses. To connect these two networks over the Internet, the external interfaces of the OpenVPN machines should be configured to use a static IP address, or be connected to another device (such as a firewall or DSL modem).



1. Generate a preset shared key.

   # openvpn --genkey --secret secrouter.key

2. Copy the file that is generated to the OpenVPN machine.

   # scp /etc/openvpn/secrouter.key 192.168.8.174:/etc/openvpn

NOTE A preshared key is stored in the directory /etc/openvpn/secrouter.key. You can use this key for testing purposes. We recommend that you create a new key dedicated for network deployment.

3. On the **OpenVPN A** machine, change the remote address in the configuration file **/etc/openvpn/tap0-br.conf**.

```
# 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
```

4. Configure the routing table in **/etc/openvpn/tap0-br.sh script**.

```
#--------------------------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--------------------------
```

5. 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”
...
```

6. Restart the bridge script file to make the bridge interface configuration take effect.

```
# /etc/openvpn/bridge restart
```

7. On the OpenVPN B machine, change the remote address in the 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
```

8. Configure the routing table in **/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 -----------------------------
```

9. 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=“192.168.8.174”
eth_netmask=“255.255.255.0”
eth_broadcast=“192.168.8.255”
#gw=“192.168.8.173”
...
```

10. Restart the bridge script file to make the bridge interface configuration take effect.

```
/etc/openvpn/bridge restart
```

NOTE You can select the cypher and authentication algorithm to use for the VPN connection. To show the list of cypher and authentication algorithms, use the following commands:

```
openvpn --show-ciphers
openvpn --show-auths
```

11. Start the OpenVPN client on the OpenVPN A and OpenVPN B machines.

```
openvpn --config /etc/openvpn/tap0-br.conf&
```

If the message **Peer Connection Initiated with 192.168.8.173:5000** appears, the connection between OpenVPN machines has been established successfully on UDP port 5000.

NOTE You can use the following command to create link symbols to start the OpenVPN service at boot time.

```
ln -sf /etc/init.d/openvpn /etc/rc2.d/S16openvpn
```

To stop the service, create the following symbolic links:

```
ln -sf /etc/init.d/openvpn /etc/rc0.d/K80openvpn
ln -sf /etc/init.d/openvpn /etc/rc6.d/K80openvpn
```

12. On each OpenVPN machine, check the routing table by typing the command `route`.

For this example, both interface **eth1** and device **tap0** 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. When the OpenVPN program detects traffic on the virtual device, the traffic is sent to the peer.

```
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
```

13. To create an indirect connection to Host B from Host A, add the following routing entry:

```
# 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, add the following routing entry:

```
# route add –net 192.168.2.0 netmask 255.255.255.0 dev eth0
```

Ping Host B from Host A by typing:

```
# ping 192.168.4.174
```

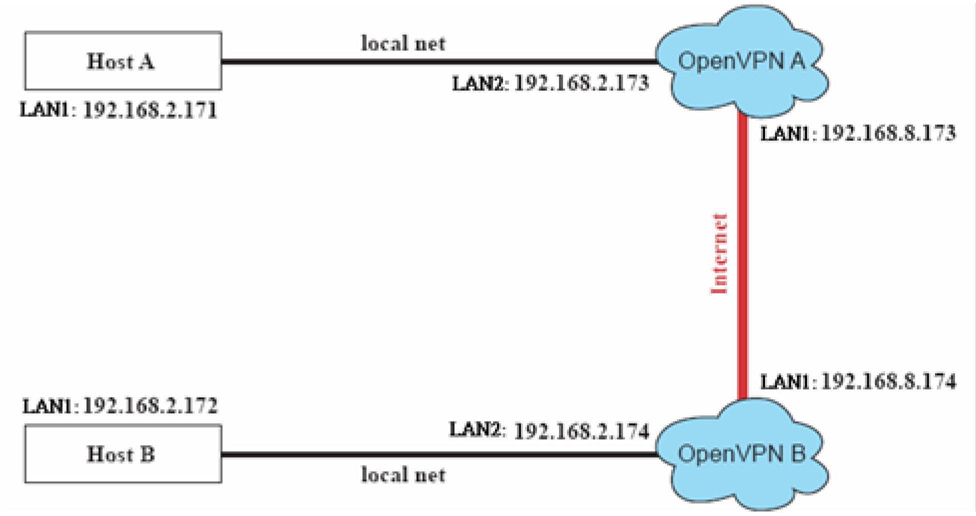
A successful ping indicates that you have created a VPN system that only allows authorized users from one internal network to access users at the remote site. For this system, all data is transmitted by UDP packets on port 5000 between OpenVPN peers.

14. To terminate the OpenVPN programs, type the command:

```
# killall -TERM openvpn
```

Ethernet Bridging for Private Networks on the Same Subnet

This section shows you how to configure Ethernet bridging for the private networks on the same subnet as shown in the following figure.



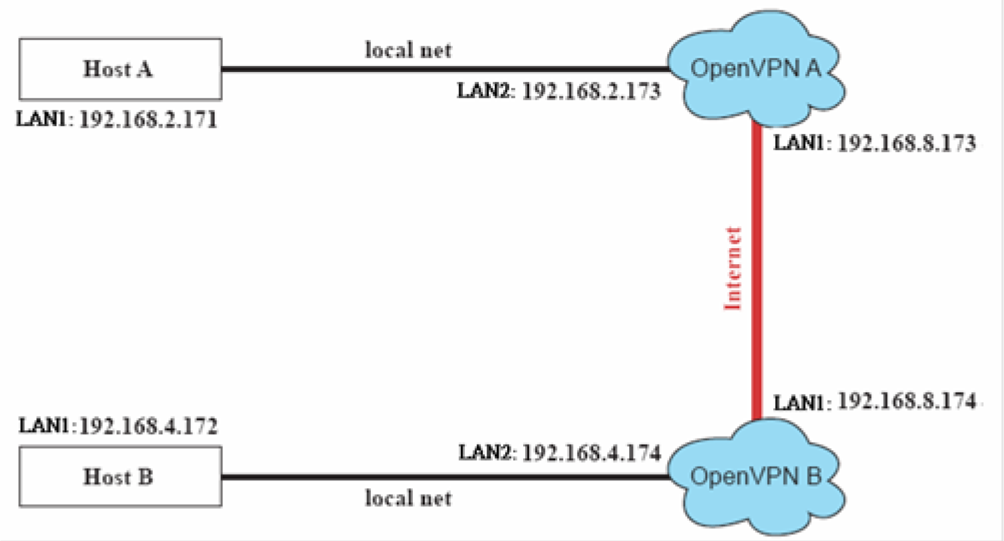
The configuration procedure is the same as the for the previous example, except that you must type "#" to comment out the line starting with "up" in the files **/etc/openvpn/tap0-br.conf** (OpenVPN A) and **/etc/openvpn/tap0-br.conf** (OpenVPN B).

The following shows an example.

```
# 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
```

Configuring IP Routing

This section shows you how to configure IP routing for the network example shown in the following figure.



1. On the OpenVPN A machine, change the remote address in configuration file **/etc/openvpn/tun.conf**.

```
# 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
-----
```

NOTE In the **tun.conf** file, the first argument of **ifconfig** is the local internal interface and the second argument is the internal interface at the remote peer.

2. Configure the routing table in script file **/etc/openvpn/tun.sh**.

```
#---------------------------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-----------------------------
```

NOTE In the **tun.sh** file, **$5** is the argument that the OpenVPN program passes to the script file. Its value is the second argument of **ifconfig** in the **tun.conf** file.

3. On the OpenVPN B machine, change the remote address in configuration file **/etc/openvpn/tun.conf**.

```
# 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
```

4. Configure the routing table in script file **/etc/openvpn/tun.sh**.

```
#--------------------------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-----------------------------
```

5. Check the routing table after you run the OpenVPN programs. Type the command **route**.

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

4

Programming Guide

The following topics are covered in this chapter:

Getting Product Serial Number

You can use the **dmidecode** command to display the product information (for example, device manufacturer and device serial number).

```
moxa@Moxa:~$ sudo dmidecode -s "baseboard-manufacturer"
MOXA
moxa@Moxa:~$ sudo dmidecode -s "baseboard-serial-number"
TACCA1000000
```

The following table shows the list of command options to display other product information.

```
  bios-vendor
  bios-version
  bios-release-date
  system-manufacturer
  system-product-name
  system-version
  system-serial-number
  system-uuid
  baseboard-manufacturer
  baseboard-product-name
  baseboard-version
  baseboard-serial-number
  baseboard-asset-tag
  chassis-manufacturer
  chassis-type
  chassis-version
  chassis-serial-number
  chassis-asset-tag
  processor-family
  processor-manufacturer
  processor-version
  processor-frequency
```

RTC (Real Time Clock)

The device node is located in **/dev/rtc**. The DA-820-LX supports standard Linux simple RTC control. You must include **<linux/rtc.h>**.

- Function: RTC_RD_TIME

```
int ioctl(fd, RTC_RD_TIME, struct rtc_time *time);
Description: read time information from the RTC. It will return the value on
argument 3.
```

- Function: RTC_SET_TIME

```
int ioctl(fd, RTC_SET_TIME, struct rtc_time *time);
Description: set RTC time. Argument 3 will be passed to RTC.
```

UART

The embedded serial ports on DA-820-LX use the general 8250 device driver. The tty device nodes are **/dev/ttyS0** and **/det/ttyS1**. The embedded serial ports support standard Linux termios control with RS-232/422/485 serial ports.

Setting the UART Mode

You can use the **setuartmode** command to configure the UART operating mode.

```
Get/set the serial port mode utility
Usage: setuartmode -n [the nth port] [-g|-s] [-h]
    Show the mx_exsi_pled information if no argument apply.
        -h: Show this information.
        -n [the nth port]: Indicate the n-th serial port.
            1 for the first serial port (default)
            2 for the second serial port.
        -g: Get the n-th serial port interface type.
        -s: Set the n-th serial port interface.
            0 for RS-232.
            1 for RS-485-2W.
            2 for RS-422/RS-485-4W.
```

The following table shows some `setuartmode` command examples.

Command Example	Description
setuartmode -n 1 -s 0	Sets the first serial port to operate in RS-232 mode.
setuartmode -n 1 -s 1	Sets the first serial port to operate inRS-485-2W mode.
setuartmode -n 1 -s 2	Sets the first serial port to operate in RS-422/485-4W mode.
setuartmode -n 1 -g	Displays the operating mode of the first serial port.
setuartmode -n 2 -s 0	Sets the second serial port to operate in RS-232 mode.
setuartmode -n 2 -s 1	Sets the second serial port to operate in RS-485-2W mode.
setuartmode -n 2 -s 2	Sets the second serial port to operate in RS-422/485 mode.
setuartmode -n 2 –g	Displays the operating mode of the second serial port.

You can install a Moxa multi-port serial board (for example, CP-102UL, CP-114UL, or CP-118U) on the DA-820 computer. The tty device nodes are **/dev/ttyMUE0**, and **/det/ttyMUE1**, etc. You can use the muestty utility to configure the UART to function in RS-232/422/485 mode.

```
Usage: muestty <operation> device
Device: The MUE series device node
Operation: -h Help
-g Get interface and terminator type
-i intf Set interface type with options below
-t value Set termination resistor with options below
intf RS232 RS-232 mode
RS422 RS-422 mode
RS4852W RS-485 2-wire mode
RS4854W RS-485 4-wire mode
value NONTERM Non termination resistor
120TERM 120ohm termination resistor
```

The following table shows some muestty command examples.

Command Example	Description
muestty –i RS422 /dev/ttyMUE2	Sets the MUE interface.
muestty –t 120TERM /dev/ttyMUE2	Sets the MUE termination resistor.

By default, the serial interface on the DA-820 is set to RS-232 mode. You can use the setinterface command to change the serial port operation mode.

setinterface device-node [interface-no]

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

Interface Number	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 operate in RS-422 mode.

```
MOXA:~# setinterface /dev/ttyM0 2
MOXA:~# setinterface /dev/ttyM0
Now setting is RS422 interface.
MOXA:~#
```

Programmable LED Indicators

There are four programmable LED indicators on the front panel of the DA-820. The programmable LED device file is located in **/dev/pled**. Each LED can be accessed via the **/dev/pled** device node.

The following table shows some examples to control the programmable LED indicators.

Command Example	Description
echo 10000000 > /dev/pled	Turns on the first LED and turns off all other LEDs.
echo 00000000 > /dev/pled	Turns off all LEDs.
echo 01000000 > /dev/pled	Turns on the second LED and turns off all other LEDs.
echo 11010000 > /dev/pled	Turns on the first and last LEDs, and turns off the second and third LEDs.

Watch Dog Timer

You can enable the Watch Dog Timer (WDT) that acts like a watchdog function. The WDT sets the system to reboot if an application does not acknowledge within the time specified (between 1 millisecond to 255 seconds).

Debian project supports a watchdog daemon to check the health of your system. If programs are no longer executed, it sets the system to perform a hard reset. The standard watchdog driver and package have been installed in the DA-820.

You can use the **insserv** command to execute the watchdog function during system startup.

```
moxa@Moxa:~$ sudo insserv -v -d watchdog
[sudo] password for moxa:
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc0.d/K01watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc1.d/K01watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc2.d/S23watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc3.d/S23watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc4.d/S23watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc5.d/S23watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc6.d/K01watchdog
insserv: creating .depend.boot
insserv: creating .depend.start
insserv: creating .depend.stop
moxa@Moxa:~$
```

Use the **ls l** command to check the runlevel of the watchdog function.

```
moxa@Moxa:~$ ls -l /etc/rc?.d/*watchdog*
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc0.d/K01watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc1.d/K01watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc2.d/S23watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc3.d/S23watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc4.d/S23watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc5.d/S23watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc6.d/K01watchdog -> ../init.d/watchdog
moxa@Moxa:~$
```

The watchdog configure file is located in **/etc/watchdog.conf**. By default, the watchdog daemon is set to check the watchdog device every 60 seconds. You can configure real-time settings for the watchdog daemon to lock itself into memory and can never be swapped out. The real-time setting prevents any delay in acknowledgement under heavy system load.

You can configure the **/etc/watchdog.conf** file to enable watchdog and specify related settings.

```
…
watchdog-device = /dev/watchdog
…
interval             = 60
realtime             = yes
priority             = -10
…
```

To remove watchdog from runlevels, use the following command.

```
moxa@Moxa:~# sudo insserv -r watchdog
```

The following command checks whether watchdog is in the runlevels.

```
moxa@Moxa:~# ls -l /etc/rc?.d/*watchdog*
ls: cannot access /etc/rc?.d/*watchdog*: No such file or directory
moxa@Moxa:~#
```

The Watchdog Device IOCTL Commands

IOCTL	**WDIOC_GETSUPPORT**
Description	This returns the support of the card itself
Input	None
Output	(struct watchdog_info *) arg
Return	On success, return 0. Otherwise, return < 0 value.
IOCTL	**WDIOC_GETSTATUS**
Description	This returns the status of the card
Input	None
Output	(int *)arg
Return	On success, return 0. Otherwise, return < 0 value.
IOCTL	**WDIOC_GETBOOTSTATUS**
Description	This returns the status of the card that was reported at bootup.
Input	None
Output	(int *)arg)
Return	On success, return 0. Otherwise, return < 0 value.
IOCTL	**WDIOC_SETOPTIONS**
Description	This lets you set the options of the card. You can either enable or disable the card this way.
Input	None
Output	(int *)arg)
Return	On success, return 0. Otherwise, return < 0 value.
IOCTL	**WDIOC_KEEPALIVE**
Description	This pings the card to tell it not to reset your computer.
Input	None
Output	None
Return	On success, return 0. Otherwise, return < 0 value.
IOCTL	**WDIOC_SETTIMEOUT**
Description	Set the watchdog timeout
Input	arg: 1 ~ 255 seconds
Output	None
Return	On success, return 0. Otherwise, return < 0 value.
IOCTL	**WDIOC_GETTIMEOUT**
Description	Get the current watchdog timeout.
Input	None
Output	arg: 1 ~ 255 seconds
Return	On success, return 0. Otherwise, return < 0 value.

Examples

The example file **watchdog-simple.c** acknowledges the watchdog every 10 seconds.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main(void)
{
        int fd = open("/dev/watchdog", O_WRONLY);
        int ret = 0;
        if (fd == -1) {
                perror("watchdog");
                exit(EXIT_FAILURE);
        }
        while (1) {
                ret = write(fd, "\0", 1);
                if (ret != 1) {
                        ret = -1;
                        break;
                }
                sleep(10);
        }
        close(fd);
        return ret;
}
```

TPM

Through the TCG Software Stack (TSS) API, TPM services provides the following features:

1. RSA key pair generation
2. RSA encryption and decryption using PKCS v1.5 and OAEP padding
3. RSA signature and verification
4. Extend data into the TPM's PCRs and log these events
5. Seal data to arbitrary PCRs
6. Random Number Generation
7. RSA key storage

For more information, see the chapter on Configuring the TMP Module or refer to the TrouSurS web site at http://trousers.sourceforge.net/

5

Optional Modules

The DA-820 comes with PCI/PCI express expansion slots that allow users to install various expansion modules. This chapter describes how to program these modules with different communication interfaces.

The following topics are covered in this chapter:

Programming Serial Modules

You can install serial expansion modules (for example, CP-102, CP-104, CP-112 or CP-118U) in the DA-820. The drivers for the expansion modules are already installed in the official DA-820-LX firmware.

After the system is started, use the **lsmod** command to check the **mxser** module.

```
moxa@Moxa:~# lsmod|more
Module                Size  Used by
…
mxser                50615  0
…
```

If the **mxser** module is loaded, the **/dev/ttyM0~/dev/ttyMn** device files are created in the system. The **/dev/ttyM0** device file is used to control serial port 0, **/dev/ttyM1** device file controls serial port 1, and similarly for the **/dev/ttyM2** device file, etc.

```
root@Moxa:~# ls -al /dev/ttyM*
crw-rw---T 1 root dialout 30,  0 Aug 14 13:37 /dev/ttyM0
crw-rw---T 1 root dialout 30,  1 Aug 14 13:37 /dev/ttyM1
crw-rw---T 1 root dialout 30, 10 Aug 14 13:37 /dev/ttyM10
crw-rw---T 1 root dialout 30, 11 Aug 14 13:37 /dev/ttyM11
crw-rw---T 1 root dialout 30, 12 Aug 14 13:37 /dev/ttyM12
crw-rw---T 1 root dialout 30, 13 Aug 14 13:37 /dev/ttyM13
crw-rw---T 1 root dialout 30, 14 Aug 14 13:37 /dev/ttyM14
crw-rw---T 1 root dialout 30, 15 Aug 14 13:37 /dev/ttyM15
crw-rw---T 1 root dialout 30,  2 Aug 14 13:37 /dev/ttyM2
crw-rw---T 1 root dialout 30,  3 Aug 14 13:37 /dev/ttyM3
crw-rw---T 1 root dialout 30,  4 Aug 14 13:37 /dev/ttyM4
crw-rw---T 1 root dialout 30,  5 Aug 14 13:37 /dev/ttyM5
crw-rw---T 1 root dialout 30,  6 Aug 14 13:37 /dev/ttyM6
crw-rw---T 1 root dialout 30,  7 Aug 14 13:37 /dev/ttyM7
crw-rw---T 1 root dialout 30,  8 Aug 14 13:37 /dev/ttyM8
crw-rw---T 1 root dialout 30,  9 Aug 14 13:37 /dev/ttyM9
```

Configuring Serial Port Mode

You can use the setinterface command to display a list of parameters and command options for serial port configuration.

```
Moxa:~# setinterface
    Usage: setinterface device-node [interface-no]
    device-node - /dev/ttyMn; n = 0,1,2,...
    interface-no    - following:
    none - to view now setting
    0 - set to RS232 interface
    1 - set to RS485-2WIRES interface
    2 - set to RS422 interface
    3 - set to RS485-4WIRES interface
Moxa:~#
```

The following lists the serial modes:

0 – Set to RS232 interface
1 - Set to RS485-2WIRES interface
2 - Set to RS422 interface
3 - Set to RS485-4WIRES interface

The following example checks the current interface setting. In this example, serial port 1 is set as an RS-485-2Wire interface. (M0 refers to port 1, M1 refers to port2, and so on)

```
Moxa: ~# setinterface /dev/ttyM0
Now setting is RS485-2WIRES interface.
```

The following example sets serial port 1 to RS-422 mode.

```
Moxa: ~# setinterface /dev/ttyM0 2
Moxa: ~# setinterface /dev/ttyM0
Now setting is RS422 interface.
```

Changing the Default Serial Mode

To change the default mode for serial interfaces, edit **/etc/udev/rules.d/96-moxa.rules**. Change the command line **RUN+="/bin/setinterface /dev/ttyM%n 0”**.

```
Moxa:~# vi /etc/udev/rules.d/96-moxa.rules
…
# Example to set the device, DA-SP08-I-DB, 0x1393:0x1180 default as 485-2W mode
interface
KERNEL=="ttyM0", RUN+="/bin/setinterface /dev/ttyM%n 1"
```

To set the default serial mode to RS-232, enter the following command.
RUN+="/bin/setinterface /dev/ttyM%n 0"

To set the default serial mode to RS-485 2-wire, enter the following command.
RUN+="/bin/setinterface /dev/ttyM%n 1"

To set the default serial mode to RS-422, enter the following command.
RUN+="/bin/setinterface /dev/ttyM%n 2"

To set the default serial mode to RS-485 4-wire, use the following command.
RUN+="/bin/setinterface /dev/ttyM%n 3"

Restart your computer.

```
Moxa:~# reboot
```

When the computer has been restarted, check if the setting has been loaded as the default value.

```
Moxa:~# setinterface /dev/ttyM0
Now setting is RS485-2WIRES interface.
Moxa:~#
```

NOTE Since the CP-102, CP-104, CP-112 and CP-118U expansion module use the same driver, mxser.ko, the /dev/ttyM* deice scanning order is pre-defined in the driver. This means the device scanning order is fixed. You have to check the order in which the expansion modules are installed in the system and verify with the scan order by /dev/ttyM*.

Programming the IRIG-B Module

You can install an IRIG-B module in the DA-820. The IRIG-B module uses the `moxa_irigb` driver. When the system boots up with the IRIG-B module installed, you can use the `lsmod` command to check if the IRIG-B module is loaded.

```
root@Moxa:~# lsmod|grep irig
moxa_irigb           12683  1
```

The following example shows that the moxa_irigb driver is loaded at startup.

```
root@Moxa:~# grep moxa_irigb /etc/modules
moxa_irigb
```

The following example shows that the IRIG-B time sync daemon is running with default settings.

```
root@Moxa:~# root@Moxa:~# ps aux|grep ServiceSyncTime
root      3078  0.0  0.1  16136  1140 ?       S    10:43   0:00
/usr/sbin/ServiceSyncTime -t 1 -o 1 -i 10
```

The following figure shows the help information on using the ServiceSyncTime command.

```
root@Moxa:~# ServiceSyncTime -h
IRIG-B time sync daemon.
Usage: ServiceSyncTime -t [interface type] -o [port in output mode] -I -w [PPS width]
-d -i [Time sync interval]
   -t [interface type]
       0 - TTL
       1 - DIFF
       default vlaue is 1
   -o [port in output mode] - Enable the IRIG1, IRIG-B module 1, in output mode
       2 - Enable IRIG-B module 1 in output mode
       default vlaue is 2
   -I - inverse the input or output signal
   -w [PPS width] - Set the wide of pulse per second in ms
       The PPS width should be 0 ~ 1000.
       default vlaue is 0
   -d - Disable time sync
       Default this daemon enables the IRIG-B time sync from source port to system time.
   -s [Time Source] - The sync source from IRIG-B decoded module 1.
       2 - IRIG-B decoded module 1
       default vlaue is 2
   -i [Time sync interval] - The time interval in seconds to sync the IRIG-B time into
system time.
       1 ~ 86400 Time sync interval. Default is 10 second.
```

The following example enables the IRIG-B time sync daemon to sync time from PORT2 (IRIG-B module 1) in TTL interface type every 10 seconds, and sets the PORT2 (IRIG-B module 1) output signal. The input and output signals are not inversed.

```
root@Moxa:~#  ServiceSyncTime -t 0 -s 2 -o 2 -i 10
```

The following example enables the IRIG-B time sync daemon to sync time from PORT2 (IRIG-B module 1) in DIFF interface type every 10 seconds, and sets the PORT2 (IRIG-B module 1) output signal. The input and output signals are not inversed.

```
root@Moxa:~#  ServiceSyncTime -t 1 -s 2 -o 2 -i 10
```

The following example enables the IRIG-B time sync daemon to sync time from PORT2 (IRIG-B module 1) in DIFF interface type every 10 seconds, and sets the PORT2 (IRIG-B module 1) output signal. Inverse the output signal if the cable connection is crossed.

```
root@Moxa:~# ServiceSyncTime -t 1 -s 2 -i 10 -I 1
```

The following example disables the IRIG-B time sync daemon.

```
root@Moxa:~# ServiceSyncTime -d
```

Configure the IRIG-B Time Sync Daemon

The IRIG-B time sync daemon is managed by the /etc/init.d/mx_irigb.sh script. The default configuration, MX_IRIGB_OPTS, is set in /etc/init.d/mx_irigb.sh.

```
root@Moxa:~# sudo vi /etc/init.d/mx_irigb.sh
...
MX_IRIGB_OPTS="-t 1 -o 2 -i 10"
...
```

After you edit the script file, restart the daemon.

```
root@Moxa:~# sudo service mx_irigb.sh restart
```

Programming the Fiber Module

You can install a fiber module in the DA-820. The fiber module uses the e1000e.ko driver that comes with the official DA-820-LX firmware. You can use the **ifconfig** command to configure this Ethernet interface.

```
Moxa:~# cat /proc/net/dev
eth0
…
eth4
eth5
Moxa:~# ifconfig eth6 192.168.9.127 up
…
Moxa:~# ifconfig eth4
eth4      Link encap:Ethernet  HWaddr 00:90:e8:00:e0:07
        inet addr:192.168.9.127  Bcast:192.168.9.255  Mask:255.255.255.0
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
        Interrupt:45 Base address:0x8000
…
```

6

Managing Disks

The DA-820-LX computers come with a software-RAID management platform designed to serve as a front-end for data acquisition and industrial control applications. This chapter describes how to configure the volume supported by the Linux operating system.

The following topics are covered in this chapter:

Configuring Software RAID Using mdadm

In Linux, you can use the mdadm utility to manage software RAID devices. The RAID volume is built in Linux, not in BIOS. mdadm, which replaces the previous utility mdctl, enables you to administer and manager multiple devices (the "md" in the name).

Creating a Software RAID Volume

The DA-820-LX computers come with two SATA disk slots. You can manage linear, RAID0, or RAID1 volumes on these two SATA disks.

1. If a RAID device is created previously, stop it and create your own RAID devices.

```
root@Moxa:~# cat /proc/mdstat
Personalities: [raid0]
md0: active raid0 sdc[0] sdb[1]
     195371008 blocks super 1.2 512k chunks
root@Moxa:~# mdadm –stop /dev/md0
root@Moxa:~# cat /proc/mdstat
Personalities : [raid0]
Unused devices: <none>
root@Moxa:~#
```

2. Unmount the disks if they are mounted.

```
root@Moxa:~# umount /media/disk1p1
root@Moxa:~# umount /media/disk2p1
root@Moxa:~# /etc/init.d/mxhtspd.sh stop
```

3. Create the partitions on the disk.

```
oot@Moxa:~# fdisk /dev/sdb1
root@Moxa:~# fdisk /dev/sdc1
root@Moxa:~# fdisk /dev/sdd1
root@Moxa:~# fdisk /dev/sde1
```

4. Create the RAID volume.

   The following figure shows the mdadm options that you can use for RAID volume creation. You can choose to create a linear mode, striping mode, or mirror mode in RAID volume.

```
-C: create
-v: verbose
-l: RAID level, options are: linear, raid0, 0, stripe, raid1, 1, mirror, raid4,
4, raid5, 5, raid6, 6, raid10, 10, multipath, mp, faulty. Obviously some of these
are synonymous.
-n: the number of disks
```

5. Create a software RAID volumn.

   The following example creates a linear mode software RAID.

```
root@Moxa:~# mdadm -Cv -llinear -n2 /dev/md0 /dev/sd{b,c}1
```

   The following example creates a striping mode software RAID 0.

```
root@Moxa:~# mdadm -Cv –l0 -n2 /dev/md0 /dev/sd{b,c}1
```

   The following example creates a mirror mode software RAID 1.

```
root@Moxa:~# mdadm -Cv –l1 -n2 /dev/md0 /dev/sd{b,c}1
```

   The following example creates a mirror mode software RAID 5.

```
root@Moxa:~# mdadm -Cv –l5 –n3 /dev/md0 /dev/sd{b,c,d}1
```

The following example creates a mirror mode software RAID 10.

```
root@Moxa:~# mdadm -Cv -l10 -n4 /dev/md0 /dev/sd{b,c,d,e}1
```

6. Check the RAID device information in /proc/mdstat.

   The following example shows the status of RADI0 and linear RAID.

```
root@Moxa:~# cat /proc/mdstat
Personalities: [raid0] [linear]
md0: active raid0 sdb1[1] sdc1[0]
     195369504 blocks super 1.2 OK rounding

Unused devices: <none>
```

   The following example shows the status of RADI1.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [linear] [raid10] [raid6] [raid5] [raid4] [raid1]
md0 : active (auto-read-only) raid1 sde1[3] sdd1[2] sdc1[1] sdb1[0]
     488252736 blocks super 1.2 [4/4] [UUUU]
       resync=PENDING

unused devices: <none>
```

   The following example shows the status of RADI5.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [linear] [raid10] [raid6] [raid5] [raid4]
md0 : active (auto-read-only) raid5 sde1[4](S) sdd1[2] sdc1[1] sdb1[0]
     1464757248 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/3] [UUU_]

unused devices: <none>
```

   The following example shows the status of RADI10.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [linear] [raid10]
md0 : active (auto-read-only) raid10 sde1[3] sdd1[2] sdc1[1] sdb1[0]
     976504832 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]
       resync=PENDING

unused devices: <none>
```

7. Format the RAID volumn.

```
root@Moxa:~# mkfs.ext4 /dev/md0
```

8. Mount the RAID volumn.

```
root@Moxa:~# mount /dev/md0 /mnt/raid
```

9. Start the RAID volume automatically for the next system startup.

   If you want to start the array automatically, edit the /etc/mdadm/mdadm.conf file.

```
root@Moxa:~# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

   The following figure shows an example.

```
DEVICE /dev/sdb1 /dev/sdc1
CREATE owner=root group=disk mode=0660 auto=yes
HOMEHOST <system>
MAILADDR your_email@xxx.com
ARRAY /dev/md0 metadata=1.2 name=Moxa:0 UUID=45ae9dbe:f30741ec:b22eff98:2dadb12d
```

10. Edit the /etc/fstab file and add the following line to mount the RAID volume.

```
/dev/md0    /mnt/raid   ext4    defaults    0   2
```

11. Unmount the root file system and reboot.

    The array should be started and mounted at /mnt/raid.

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

Displaying Software RAID Status

You can view the software RAID status by displaying /proc/mdstat.

The following example shows the status while the array is running.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [linear]
md0 : active linear sdb1[1] sdc1[0]
     23436724 blocks super 1.2 0k rounding
```

The following example shows the status while the array is not running.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [linear]
unused devices: <none>
```

Replacing a Failed Disk

If the array is running in mirror mode and one of the disks fails, you should replace the failed disk with a new one.

1. Check the status of the RAID array. In the following example, sdb1[0](F) indicates that the sdb disk is failed.

```
md1 : active raid1 sdb1[1] sdc1[0](F)
17920384 blocks [2/1] [_U]
```

   You can simulate a disk failure by entering the following command.

```
root@Moxa:~# mdadm --manage /dev/md0 --fail /dev/sdb1
mdadm: set /dev/sdb1 faulty in /dev/md0
root@Moxa:~# sync
```

2. Remove a failed disk from the RAID array.

```
root@Moxa:~# mdadm -r /dev/md0 /dev/sdb1
mdadm: hot removed /dev/sdb1 from /dev/md0
```

3. Replace the failed disk drive and add the drive volumn into the RAID array.

```
root@Moxa:~# mdadm -a /dev/md0 /dev/sdb1
```

4. Display mdstat to check the RAID array status. The following figure shows that the RAID array has automatically recovered.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdb1[0] sdc1[1]
     7806522 blocks super 1.2 [2/1] [_U]
     [==>..................]  recovery = 10.6% (831488/7806522) finish=0.9min
speed=118784K/sec

unused devices: <n
```

7

Configuring the TPM Module

DA-820-LX Series integrate a Trusted Platform Module (TPM) that provides added protection in the system.

The following topics are covered in this chapter:

- ❒ **Trusted Platform Module (TPM) and TrouSerS**
- ❒ **Enabling the TPM**
- ❒ **Starting TPM Services**
- ❒ **Initializing the TPM**
- ❒ **Getting the Public Endorsement Key**
- ❒ **Sealing and Unsealing Data**

Trusted Platform Module (TPM) and TrouSerS

TPM is a microcontroller that can securely store information such as passwords, certificates, or encryption keys which are used to authenticate the platform. TMP provides hardware-based data protection because the private key used to protect the data is never exposed in the clear outside of the TPM's own internal memory area.

A TPM can also be used to store platform measurements to help ensure a trusted platform. Data can also be protected by these measurements as well as requiring the platform to be in the same configuration to access the data as when the data was first protected.

The TPM specification was written by a computer industry consortium called the Trusted Computing Group (TCG). TrouSerS implements the TCG Software Stack (TSS) that contains the tcsd daemon and the TPM tool for you to access to and communicate with the TPM. These packages are pre-installed on the DA-820-LX.

The following table lists the supported TPM tool commands.

Command	**Description**
tpm_changeownerauth	Change the authorization data associated with the owner or SRK.
tpm_clear	Return the TPM to the default state (unowned, disabled, inactive).
tpm_createek	Create an Endoresement Key pair in the TPM.
tpm_getpubek	Display the public portion of the Endoresement Key in the TPM.
tpm_resetdalock	Reset the dictionary attack lock for the user (requires owner authentication).
tpm_restrictpubek	Restrict the ability to display the public portion of the Endorsement Key to the owner.
tpm_revokeek	Revoke the Endorsement Key pair of the TPM.
tpm_sealdata	Seal input data to the system TPM.
tpm_selftest	Request the TPM to perform selftest and report.
tpm_setactive	Change the TPM active state.
tpm_setclearable	Disable the TPM clear operation.
tpm_setenable	Change the TPM enable state.
tpm_setoperatorauth	Set the operator authorization value in the TPM.
tpm_setownable	Change if the TPM allows tpm_takeownership operation.
tpm_setpresence	Change the TPM physical presence states or settings.
tpm_takeownership	Set up an owner on the TPM.
tpm_version	Display the TPM version and manufacturer information.

The following table lists the PKCS#11 data management commands of the TPM tool.

Command	**Description**
tpmtoken_import	Import an X.590 certificate and/or an RSA key pair into the user’s TPM PKCS#11 data store.
tpmtoken_init	Initialize the user’s TPM PKCS#11 data store.
tpmtoken_objects	Display the objects in the user’s TPM PKCS#11 data store.
tpmtoken_protect	Encrypt or decrypt data using a symmetric key stored in the user’s TPM PKCS#11 data store.
tpmtoken_setpasswd	Change the passwords associated with the user’s TPM PKCS#11 data store.

NOTE Access following links for more information about TPM and TrouSerS TPM specification:

http://www.trustedcomputinggroup.org/resources/tpm_main_specification
http://trousers.sourceforge.net/
http://ibmswtpm.sourceforge.net/tpm_tss.html

Enabling the TPM

To start using the TPM on the DA-820, you must first enable the TPM function from boot loader and start the related services in the system.

1. Turn on the Da-820 and press **<F2>** to enter the BIOS configuration screen.
2. Select **Security > TPM Operation > Enable and Activate**.
3. Select **TPM Force Clear> Enabled**.
4. Restart the DA-820 to make the changes take effect.

Starting TPM Services

After you enable the TPM, you can start the **trousers** and **opencryptoki** services.

The following figure shows the command example. For information on using the insserv and /etc/init.d/command, see the Enabling and Disabling Daemons section.

```
moxa@Moxa:~$ sudo insserv -d trousers
moxa@Moxa:~$ sudo insserv -d opencryptoki
moxa@Moxa:~$ sudo /etc/init.d/trousers start
moxa@Moxa:~$ sudo /etc/init.d/opencryptoki start
```

The following command example shows the TMP version.

```
moxa@Moxa:~$ sudo tpm_version
 TPM 1.2 Version Info:
 Chip Version: 1.2.3.17
 Spec Level: 2
 Errata Revision: 2
 TPM Vendor ID: IFX
 Vendor Specific data: 03110008 00
 TPM Version: 01010000
 Manufacturer Info: 49465800
```

NOTE Before you enter a TMP command, make sure that the TMP daemon is running; otherwise, the following error message appears.
Tspi_Context_Connect failed: 0x00003011 - layer=tsp, code=0011 (17), Communication failure

Initializing the TPM

The first step to using the TPM is to configure ownership settings.

Enter the **tmp_takeownership** command and enter the owner password and SRK password.

```
moxa@Moxa:~$ sudo tpm_takeownership
Enter owner password:
Confirm password:
Enter SRK password:
Confirm password:
```

Enter owner password and SRK password twice as it requested. Notice that owner and SRK passwords which are very important and must not be lost.

NOTE You can configure the ownership settings once. You must remember the passwords you configure and store them in a secure location.

If the following error message appears, the endorsement key is not set. Enter the sudo tpm_createek command to create an endorsement key.
Tspi_TPM_GetPubEndorsementKey failed: 0x00000023 - layer=tpm, code=0023 (35), No EKlease make sure …..

Getting the Public Endorsement Key

The endorsement key is typically a 2,048-bit RSA public and private key pair, which is created randomly on the chip at the time of manufacturing and cannot be changed. The private key is stored on the chip, while the public key is used for verification and for encryption of sensitive data sent to the chip.

The following command example displays the public portion of the endorsement key.

```
moxa@Moxa:~$ sudo tpm_getpubek
Tspi_TPM_GetPubEndorsementKey failed: 0x00000008 - layer=tpm, code=0008 (8), The TPM target command has been disabled
Enter owner password:
Public Endorsement Key:
Version: 01010000
Usage: 0x0002 (Unknown)
Flags: 0x00000000 (!VOLATILE, !MIGRATABLE, !REDIRECTION)
AuthUsage: 0x00 (Never)
Algorithm: 0x00000020 (Unknown)
Encryption Scheme: 0x00000012 (Unknown)
Signature Scheme: 0x00000010 (Unknown)
Public Key:
 b1000e32 269ee2bd f2114775 dd553e8a b9bac458 cfd52496 b6dd590b 776e2fd1
 4a519f1c e1fe6085 d8365f02 261bc6f0 e1e7f2e0 833da920 970cd588 d1e6939e
 3f35a8a8 251f298d 78c46e34 d68ef7cc 7a685d9e baf7f6e5 e3bcc303 163e9e67
 395426dc 39c90b37 9aa17f55 6fbea49e 0a76fc01 cafd9062 772112c4 c9207e6c
 ebda664c 7a6cbda8 301dcc4c 67dc8f03 9ea8993a 1f9068ec 9757ec8e 26b4c6e2
 87e30470 6fbf4ae3 3e32b5b7 dfe55dfc 4da3012d b6a600bb d7eeed48 99c118b5
 4950c2f5 1527c78f 12dfcea7 d9dfdc8a 10cd442a d3f17173 784a69c7 9689c822
 f364af90 2802bfcd 5a1227c2 3c7d02b0 e7e804a3 abe8034b 3584c529 1265a881
```

Sealing and Unsealing Data

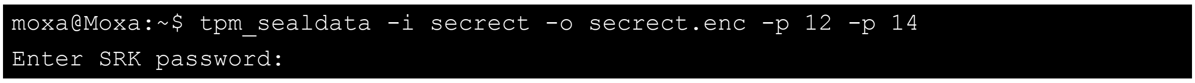
You can use the **tpm_sealdata** and **tpm_unsealdata** commands to seal and unseal sensitive data.

The following table describes the command options.

Parameter Option	Description
-i, --infile FILE	Filename containing the key to seal/unseal. The default is STDIN.
-o, --outfile FILE	Filename to write sealed/unseal key to. The default is STDOUT.
-p, --pcr NUMBER	PCR to seal data to. The default is none. This option can be specified multiple times to choose more than one PCR.

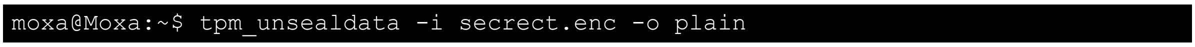
The **tpm_sealdata** command retrieves random data from the TPM. To do this, the **tpmGetRandom** function invokes the **Tspi_TPM_GetRandom()** method of the TPM class. Then, the **tpm_sealdata** command sets the SRK policy using the Policy and Context classes. The next functions build an RSA key object that will be created by the TPM. Then, an RSA key is created and loaded. The subsequent functions build an encrypted data object that will hold the encrypted version of the symmetric key. The final functions encrypt the given data and seal it to the symmetric key. It is possible to invoke this command with several command line parameters.

The following figure shows the tpm_sealdata command example.



```
moxa@Moxa:~$ tpm_sealdata -i secrect -o secrect.enc -p 12 -p 14
Enter SRK password:
```

The following figure shows the tpm_unsealdata command example.



```
moxa@Moxa:~$ tpm_unsealdata -i secrect.enc -o plain
```

8

VMWare ESXi 5.x

The DA-820-LX computers come with 64-bit CPU and hardware virtualization support that allow you to use the DA-820 as a VMWare ESXi virtualization server.

This chapter describes how to manage the peripherals on the VMWare ESXi host from a VMWare ESXi guest system.

The following topics are covered in this chapter:

- **About VMWare ESXi Peripheral Control**
- **Installing the VMWare ESXi Host Driver**
  - Verifying Package Installation
  - Removing a Package
- **Installing the VMWare ESXi VMCI Server**
- **VMWare VMCI Client Example**
- **Device Control Layer and API definitions**
- **VMCI Utility Example Code**
- **Compiling the VMCI Utility Example in Linux**
- **Compiling the VMCI Utility Example in Windows**

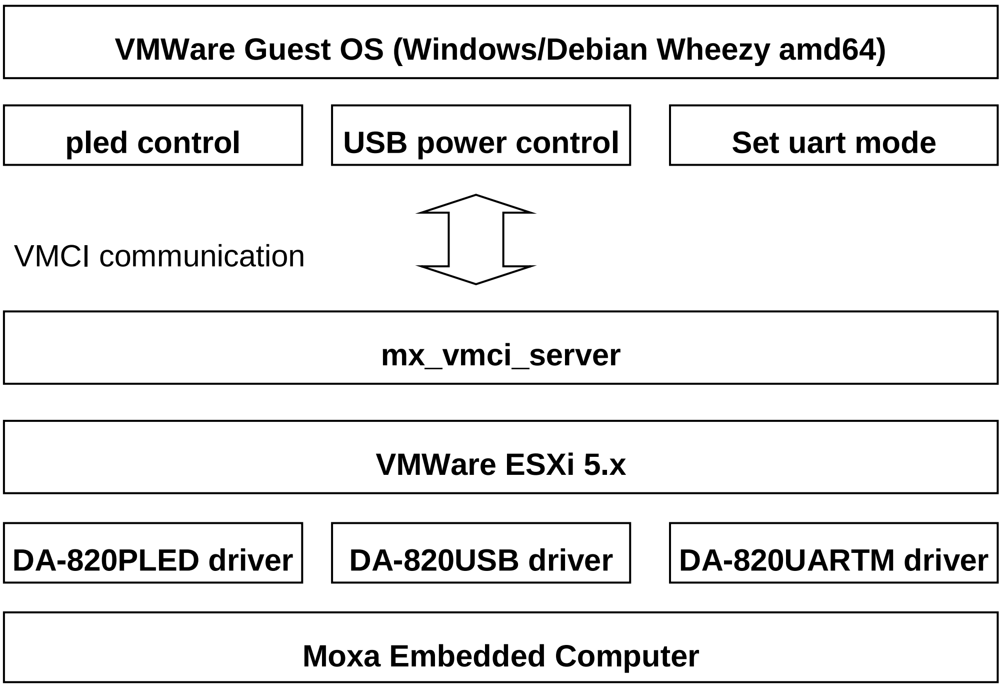
About VMWare ESXi Peripheral Control

VMWare ESXi is a free bare-metal hypervisor that virtualizes servers to enable you to consolidate your applications on a system. User can install a Windows or Linux guest operating system running on VMWare ESXi.

VMWare can access most PCI cards via PCI pass-through but not for some I/O port devices. For these I/O port devices, you can use VMCI communication to control the physical device on VMWare EXSi 5.x.

The following figure shows an overview of the communication between the VMWare guest system and VMWare ESXi 5.x. In this example, a VMCI server and mx_vmci_server running on VMWare ESXi 5.x are used to communicate with the VMCI client (mx_exsi_pled. mx_exsi_usb_power and mx_exsi_set_uart8250_mode).

The client and server use VMWare VMCI socket for communication. The VMCi client open source codes contain the VMCI client utility, mx_exsi_pled.c, mx_exsi_usb_power.c, and mx_exsi_set_uart8250.c. You can reference this code to develop your own application.



Installing the VMWare ESXi Host Driver

To control the non-PCI based peripherals, you must install the VMWare ESXi host drivers using the esxcli utility.

1. Set the system in maintenance mode.

```
# esxcli system maintenanceMode set -e true -t 0
```

2. Change the host acceptance level to CommunitySupported.

```
# esxcli software acceptance set --level=CommunitySupported
```

3. Install the programmable LED, USB power control, and UART mode driver packages.
4. Install the programmable LED package.

```
# esxcli software vib install -v /tmp/DA820PLED.vib
```

5. Install the USB power control package.

```
# esxcli software vib install -v /tmp/DA820USB.vib
```

6. Install the UART mode control package driver.

```
# esxcli software vib install -v /tmp/DA820UARTM.vib
```

7. Restart VMWare ESXi.

Verifying Package Installation

After you install all the packages, you can verify that the packages have been installed properly.

The following command example verifies the DA820PLED package installation.

```
# esxcli software vib list|grep DA820PLED
```

The following command example verifies the DA820USB package installation

```
# esxcli software vib list|grep DA820USB
```

The following command example verifies the DA820UARTM package installation

```
# esxcli software vib list|grep DA820UARTM
```

Removing a Package

You can use the **esxcli** command to remove a package from VMWare ESXi.

The following command example uninstalls the DA820PLED package.

```
# esxcli software vib remove -n DA820PLED
```

NOTE To make the changes take effect, you must restart VMWare ESXi after you uninstall a package.

Installing the VMWare ESXi VMCI Server

On VMWare ESXi, DA820VMCIS.vib is the VMCI server. You can use the **esxcli** command with the **-f** option to force install the **DA820VMCIS.vib** package. The following figure shows a command example.

```
# esxcli software vib install -v /tmp/DA820VMCI.vib  -f
Installation Result
   Message: Operation finished successfully.
   Reboot Required: false
   VIBs Installed: Moxa_bootbank_DA820VMCI_1.0
   VIBs Removed:
   VIBs Skipped:
```

Then, restart VMWare ESXi to make the changes take effect. After the system starts up, check that **mx_vmci_server** is running for the VMCI client utility to connect and control programmable LED, USB power on the front and rear panels, or UART mode.

```
# ps |grep mx_vmci_server
34564 34564 mx_vmci_server      /usr/lib/vmware/misc/bin/mx_vmci_server
```

VMWare VMCI Client Example

The following figure shows sample code in the MCI_Client file that controls the programmable LED, USB power on the front and rear panels, and UART mode.

```
mx_exsi_pled.c: The VMCI client utility for controling the programmable LED.
mx_exsi_usb_power.c: The VMCI client utility for controling the front/rear end USB
power.
mx_exsi_set_uart8250_mode.c: The VMCI client utility for setting the UART
RS-232/422/485 mode.
device_control.h: Declare the device control API, such as device_open(),
device_close(), ... .
device_control.c: Implement the device control API, such as device_open(),
device_close(), ... .
front_end_usb_power_ctrl.h: Declare the front end USB power control API.
front_end_usb_power_ctrl.c: Implement the front end USB power control API.
rear_end_usb_power_ctrl.h: Declare the rear end USB power control API.
rear_end_usb_power_ctrl.c: Implement the rear end USB power control API.
pled_ctrl.h: Declare the programmable LED control API.
pled_ctrl.c: Implement the programmable LED control API.
```

Device Control Layer and API definitions

This section includes the definition for the device_contrl function to define a virtual layer for controlling the programming LED, USB-powered device on the front and rear panels.

int device_open (char* type)

```
/*
* @brief open a device by given information
* @param type the device name (pled, front_end_usb_power,
*         rear_end_usb_power, uart8250_mode,...)
* @return
*    > 0: file descriptor;
*    <= 0 : Error
*/
/* pled */
int fd_pled = device_open("pled");
/* USB power */
int fd_usb_power_front = device_open("front_end_usb_power");
int fd_usb_power_rear = device_open("rear_end_usb_power");
/* uart8250_mode */
int fd_uart = device_open("uart8250_mode");
void device_close (int file_descriptor)
/*
* @brief close a device
* @param The file descriptor as an unique identifier return from device_open
* @return void
*/
/* pled */
device_close(fd_pled);
/* USB power */
device_close(fd_front_end_usb_power);
device_close(fd_rear_end_usb_power);
/* uart8250_mode */
```

```
device_close(fd_uart);
int device_list (int file_descriptor);
/*
* @brief list out the available devices in the given address
* @param The file descriptor as an unique identifier return from device_open
* @return number of device,
*   > 0   : The available number of devices. The device index starts from 1.
*   <= 0 : Error occurred.
*/
/* pled
* example of the devices may be:
* file_descriptor = fd_pled
* the return num should be 8 of DA-820-LX
*/
int num = device_list(fd_pled);
/* front_end_usb_power
* example of the devices may be:
* file_descriptor = fd_front_end_usb_power
* the return num should be 1
*/
int num = device_list(fd_front_end_usb_power);
/* rear_end_usb_power
* example of the devices may be:
* file_descriptor = fd_rear_end_usb_power
* the return num should be 1
*/
int num = device_list(fd_rear_end_usb_power);
/* uart8250_mode
* example of the devices may be:
* file_descriptor = fd_uart
* the return num should be 2
*/
int num = device_list(fd_uart);
int device_get (int file_descriptor, int index, int &value);
/*
* @brief get some data of the specified device
* @param The file descriptor as an unique identifier return from device_open
* @param the index of the corresponding device.
* @param the return from device_get()
* @return
*     >= 0: Success;
*     < 0 : Error
*/
/* pled */
int value;
/* The first LED index start from 1 */
int res = device_get(fd_pled, 1, &value);
/* USB power */
int on_off;/* on: 1; off: 0 */
int res = device_get(fd_front_end_usb_power, 1, &on_off);     /* Front */
int res = device_get(fd_rear_end_usb_power, 1, &on_off);      /* rear   */
/* UART mode */
/* 1 for the first serial port; 2 for the second serial port */
/* mode: 0 - RS232; 1 - RS485-2WIRES; 2 - RS422/RS485-4WIRES */
int res = device_get(fd_pled, 1, &mode);
int device_set (int file_descriptor, int index, int value);
/*
* @brief set some data of the specified device
* @param The file descriptor as an unique identifier return from device_open
* @param the index of the corresponding device.
* @param the value to be set to the device
* @return
*     >= 0: Success;
*     < 0 : Error
*/
/* pled */
int on_off=1;   /* on: 1; off: 0 */
int res = device_set(fd_pled, 1, on_off);
/* USB power front end */
int on_off=1;   /* on: 1; off: 0 */
int res = device_set(fd_front_end_usb_power, 1, on_off);
int on_off=0;   /* on: 1; off: 0 */
int res = device_set(fd_rear_end_usb_power, 1, on_off);
/* UART mode */
/* 1 for the first serial port; 2 for the second serial port */
/* mode: 0 - RS232; 1 - RS485-2WIRES; 2 - RS422/RS485-4WIRES */
int result = device_set(fd, 1, mode);
int device_read (char* type, char* data, size_t size);
/*
* @brief read data from the specified device
* @param The file descriptor as an unique identifier return from device_open
* @param data read from the device.
* @param size the size of the data read from the device.
* @return
*    0    : End of file
*    >    : The number of data read from the device
*    <0   : Error
*/
/* pled */
unsigned char data;
int ret = device_read(fd_pled, &data, sizeof(data));
/* Front end and rear end USB power control */
unsigned char data;
/* Front end USB power */
int ret = device_read(fd_front_end_usb_power, &data, sizeof(data));
/* rear end USB power */
int ret = device_read(fd_rear_end_usb_power, &data, sizeof(data));
/* UART mode   */
/* device_read() for UART mode is not supported now */
int device_write (char* type, char* data, size_t size);
/*
* @brief write data to the specified device
* @param The file descriptor as an unique identifier return from device_open
* @param data the data written to the device.
* @param size the size of the data read from the device.
* @return
*    0    : End of file
*    > 0 : The number of data written to the device
*    < 0 : Error
*/
```

```
/* pled */
unsigned char data = "11111111";
int res = device_write(fd_pled, &data, sizeof(data));
/* USB power */
unsigned char data = "1";
/* The front end USB power control */
int res = device_write(fd_front_end_usb_power, &data, sizeof(data));
/* The rear end USB power control */
int res = device_write(fd_rear_end_usb_power, &data, sizeof(data));
/* UART mode   */
/* device_write() for UART mode is not supported now */
```

VMCI Utility Example Code

The **vmci_example.tar.gz** package contains the **mx_esxi_pled.c**, **mx_esxi_set_uart8250_mode.c**, and **mx_esxi_usb_power.c** source files.

In this example, **mx_esxi_pled.c** uses the following functions to control the programmable LED:

```
device_open()
device_get()
device_set()
device_write()
device_read()
```

The following shows the example code.

```
#include <sys/types.h>
/*
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#ifndef WIN32
#include <sys/un.h>
#include <unistd.h>
#endif
#include <errno.h>
#include "vmci_sockets.h"
#include "device_control.h"      /* Define the packaet format */
void usage(char *name) {
        printf("Get/set the programmable LED utility\n");
        printf("Usage: %s -l -n [-r|-w] [-g|-s] [-h]\n", name);
        printf("      Show the mx_exsi_pled information if no argument apply.\n");
        printf("      -h: Show this information.\n");
        printf("      -l: List the number of LEDs.\n");
        printf("      -n: Indicate the n-the LED.\n");
        printf("      -r: Read the LED bitmap.\n");
        printf("      -w: Write the bitmap to the LED.\n");
        printf("      -g: Get the LED port value.\n");
        printf("      -s: Set the value to the LED port.\n");
}
#ifdef WIN32
```

```
extern int
getopt(int nargc, char *const *nargv, const char *ostr);
#endif
extern int optind, opterr, optopt;
extern char *optarg;
/* entry point */
int main(int argc, char *argv[])
{
        int fd, result;
        int num;
        int value = 0;
        int nth = 0;
        int bWrite = 0;
        int bRead = 0;
        int bSet = 0;
        int bGet = 0;
        int bList = 0;
        char led_bitmap[]="00000000";
        char optstring[] = "hn:s:gw:rl";
        char c;
        if ( argc == 1 ) {
                usage(argv[0]);
                return 0;
        }
        while ((c = getopt(argc, argv, optstring)) != -1)
                switch (c) {
                case 'h':
                        usage(argv[0]);
                        return 0;
                case 'n':
                        nth=atoi(optarg);
                        if ( nth <=0 || nth >8 ) {
                                printf(" nth:%d is not in 1 and 8\n", nth);
                                return 0;
                        }
                        break;
                case 's':
                        bSet = 1;
                        value=atoi(optarg);
                        break;
                case 'g':
                        bGet = 1;
                        break;
                case 'w':
                        bWrite = 1;
                        strcpy(led_bitmap, optarg);
                        break;
                case 'r':
                        bRead = 1;
                        break;
                case 'l':
                        bList = 1;
                        break;
                case '?':
                        printf("Invalid option\n");
                        usage(argv[0]);
                        return 0;
                default:
                        usage(argv[0]);
                        return 0;
                }
        fd = device_open("pled");
        if ( fd < 0 ) {
                printf("device_open() fail\n");
                goto main_close;
        }

        if ( ( nth >= 1 && nth <= 8 ) && bGet == 0 && bSet == 0 ) {
                printf("The -n option should be used with -g or -s\n");
                printf("EX: To set the first led on by, `mx_exsi_pled -n 1 -s 1`\n");
                goto main_close;
        }
        else if ( bList == 1 ) {
                num = device_list(fd);
                if ( num < 0 ) {
                        printf("device_list() fail\n");
                        goto main_close;
                }
                printf("pled number:%d\n", num);
        }
        else if ( bGet == 1 ) {
                result = device_get(fd, nth, &value);
                printf("Get pled[%d] value %d\n", nth, value);
                if ( result < 0 ) {
                        printf("device_list() fail\n");
                        goto main_close;
                }
        }
        else if ( bSet == 1 ) {
                if ( value == 0 ) {
                        printf("Turn off the LED, %d\n", nth);
                        result = device_set(fd, nth, 0);
                }
                else {
                        printf("Turn on the LED, %d\n", nth);
                        result = device_set(fd, nth, 1);
                }
                if ( result < 0 ) {
                        printf("device_set() fail\n");
                        goto main_close;
                }
        }
        else if ( bRead == 1 ) {
                result = device_read(fd, led_bitmap, sizeof(led_bitmap));
                if ( result < 0 ) {
                        printf("device_read() fail, result:%d\n", result);
                        goto main_close;
                }
                printf(" The led_bitmap is %s\n", led_bitmap);
        }
```

```
        else if ( bWrite == 1 ) {
                /* The size of led_bitmap should include the '\0' or '\n' */
                result = device_write(fd, led_bitmap, strlen(led_bitmap)+1);
                if ( result < 0 ) {
                        printf("device_write() fail\n");
                        goto main_close;
                }
                printf("Write the LED bitmap:%s\n", led_bitmap);
        }
main_close:
        device_close(fd);
        return 0 ;
}
```

Compiling the VMCI Utility Example in Linux

To compile the VMCI client utility in Linux, enter the commands shown in the following figure to uncompress the vmci_example.tar.gz file and compile the VMCI client.

```
root@Debian:~# tar xzvf vmci_example.tar.gz
root@Debian:~# cd vmci_example
root@Debian:~# make
```

The system creates the mx_esxi_pled, mx_esxi_set_uart8250_mode, and mx_esxi_usb_power files in the folder src/vmci_client/.

To control the programmable LED on VMWare ESXi 5.x, use the **mx_esxi_pled** command.

```
root@Debian:~# root@Debian7ESXi:/tmp# ./mx_exsi_pled
Get/set the programmable LED utility
Usage: ./mx_exsi_pled -l -n [-r|-w] [-g|-s] [-h]
    Show the mx_exsi_pled information if no argument apply.
    -h: Show this information.
    -l: List the number of LEDs.
    -n: Indicate the n-the LED.
    -r: Read the LED bitmap.
    -w: Write the bitmap to the LED.
    -g: Get the LED port value.
    -s: Set the value to the LED port.
```

The following table describes some **mx_esxi_pled** command examples.

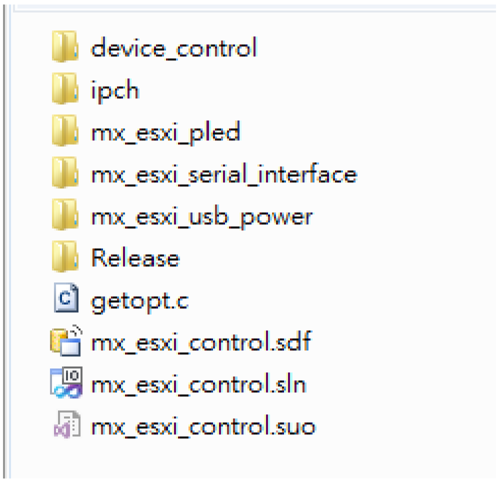
Command Example	Description
./mx_exsi_pled -l	Displays the number of LEDs.
./mx_exsi_pled -w "11110000"	Turns on LEDs 1, 2, 3, and 4. Turns off LEDs 5, 6, 7, and 8.
./mx_exsi_pled -w "10101010"	Turns on LEDs 1, 3, 5, and 7. Turns off LEDs 2, 4, 6, and 8.
./mx_exsi_pled -n 1 -s 1	Turns on LED 1.
./mx_exsi_pled -n 8 -s 1	Turns on LED 8.
./mx_exsi_pled -n 8 -s 0	Turns off LED 8.

Compiling the VMCI Utility Example in Windows

This section shows you how to compile the VMCO client in Windows.

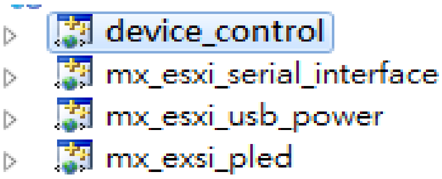
NOTE Before you start, make sure that you have Visual Studio 2010 installed in Windows.

1. Unzip the **vmci_example.tar.gz** file.
2. Double-click **mx_esxi_control.sln** to open the project.

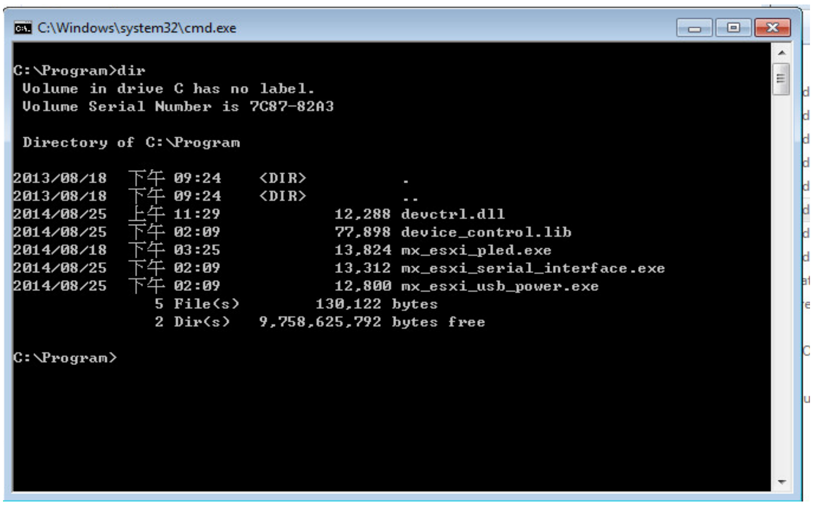


Compile the following files to generate the executable files:

mx_esxi_pled
mx_esxi_serial_interface_mode
mx_esxi_usb_power

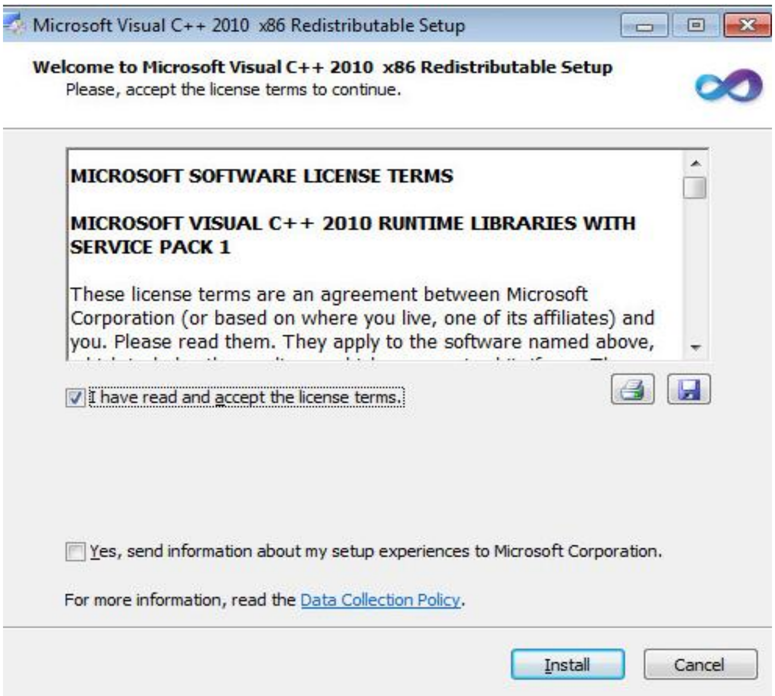


3. Copy the following files to you virtual machine:
   - devctrl.dll
   - mx_esxi_pled.exe
   - mx_esxi_serial_interface.exe
   - mx_esxi_usb_power.exe



```
C:\Program>dir
 Volume in drive C has no label.
 Volume Serial Number is 7C87-82A3

 Directory of C:\Program

2013/08/18  下午 09:24    <DIR>          .
2013/08/18  下午 09:24    <DIR>          ..
2014/08/25  上午 11:29            12,288 devctrl.dll
2014/08/25  下午 02:09            77,898 device_control.lib
2014/08/18  下午 03:25            13,824 mx_esxi_pled.exe
2014/08/25  下午 02:09            13,312 mx_esxi_serial_interface.exe
2014/08/25  下午 02:09            12,800 mx_esxi_usb_power.exe
               5 File(s)        130,122 bytes
               2 Dir(s)   9,758,625,792 bytes free

C:\Program>
```

4. Install the Visual Studio 2010 redistribute package(x86), if it is not already installed.



Welcome to Microsoft Visual C++ 2010 x86 Redistributable Setup
Please, accept the license terms to continue.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL C++ 2010 RUNTIME LIBRARIES WITH SERVICE PACK 1

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above,

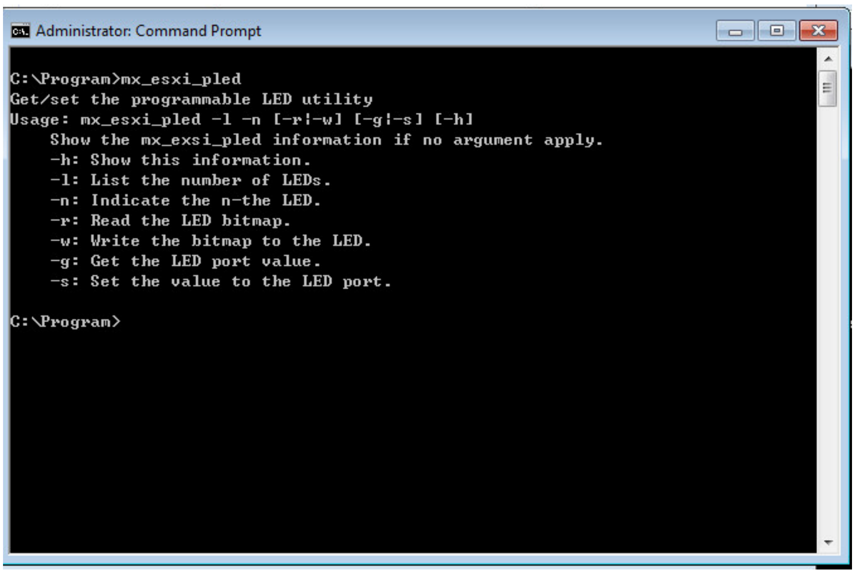
☑ I have read and accept the license terms.

☐ Yes, send information about my setup experiences to Microsoft Corporation.

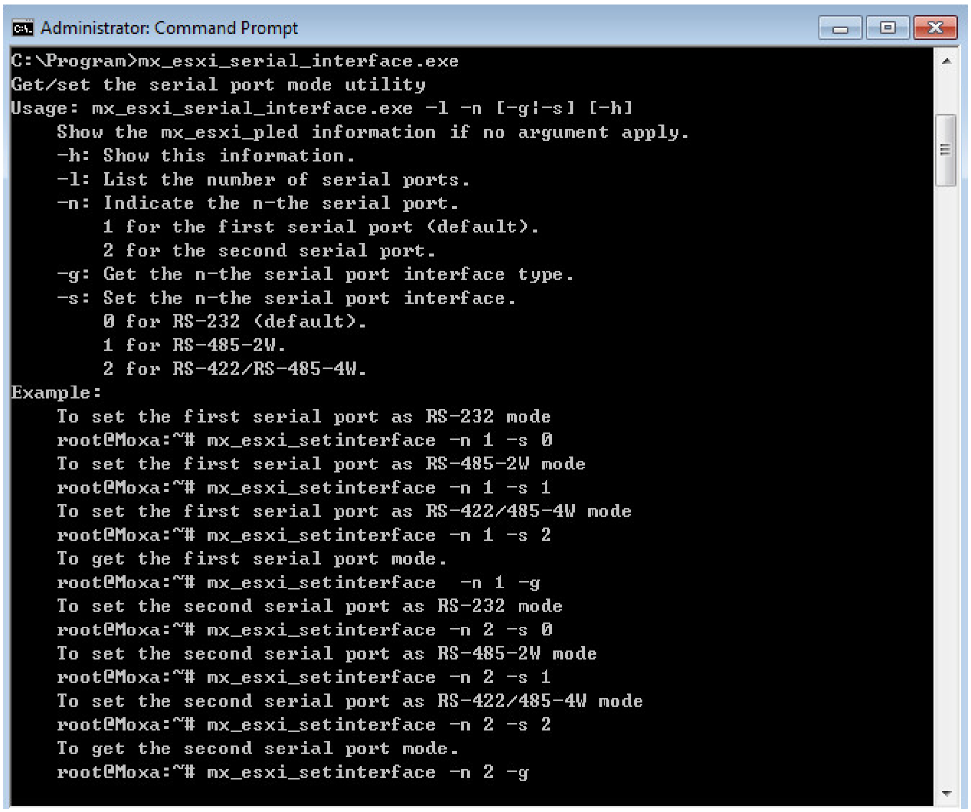
For more information, read the Data Collection Policy.

You can run the **mx_esxi_pled** command to control the programmable LEDs, serial interface, and USB power on VMWare ESXi 5.x.

The following figures show the command examples.



```
C:\Program>mx_esxi_pled
Get/set the programmable LED utility
Usage: mx_esxi_pled -l -n [-r|-w] [-g|-s] [-h]
    Show the mx_exsi_pled information if no argument apply.
    -h: Show this information.
    -l: List the number of LEDs.
    -n: Indicate the n-the LED.
    -r: Read the LED bitmap.
    -w: Write the bitmap to the LED.
    -g: Get the LED port value.
    -s: Set the value to the LED port.

C:\Program>
```



```
C:\Program>mx_esxi_serial_interface.exe
Get/set the serial port mode utility
Usage: mx_esxi_serial_interface.exe -l -n [-g|-s] [-h]
    Show the mx_esxi_pled information if no argument apply.
    -h: Show this information.
    -l: List the number of serial ports.
    -n: Indicate the n-the serial port.
        1 for the first serial port (default).
        2 for the second serial port.
    -g: Get the n-the serial port interface type.
    -s: Set the n-the serial port interface.
        0 for RS-232 (default).
        1 for RS-485-2W.
        2 for RS-422/RS-485-4W.
Example:
    To set the first serial port as RS-232 mode
    root@Moxa:~# mx_esxi_setinterface -n 1 -s 0
    To set the first serial port as RS-485-2W mode
    root@Moxa:~# mx_esxi_setinterface -n 1 -s 1
    To set the first serial port as RS-422/485-4W mode
    root@Moxa:~# mx_esxi_setinterface -n 1 -s 2
    To get the first serial port mode.
    root@Moxa:~# mx_esxi_setinterface  -n 1 -g
    To set the second serial port as RS-232 mode
    root@Moxa:~# mx_esxi_setinterface -n 2 -s 0
    To set the second serial port as RS-485-2W mode
    root@Moxa:~# mx_esxi_setinterface -n 2 -s 1
    To set the second serial port as RS-422/485-4W mode
    root@Moxa:~# mx_esxi_setinterface -n 2 -s 2
    To get the second serial port mode.
    root@Moxa:~# mx_esxi_setinterface -n 2 -g
```

9

System Recovery

The DA-820-LX ready-to-run embedded computers are an embedded Linux platform. This chapter describes the recovery process.

The following topics are covered in this chapter:

- ❒ **Overview**
- ❒ **Setting Up the System Recovery Environment**
- ❒ **Recovering from the Factory Default Image**
  - ➢ Step 1: Preparing the USB Drive
  - ➢ Step 2: Setting the BIOS to Boot from the USB Drive
  - ➢ Step 3: Performing a System Recovery
  - ➢ Step 4: Resetting the BIOS
- ❒ **Creating a Custom System Image**

Overview

This section describes the recovery process in the event of a system failure.

You can recover the system using one of the following:

- Factory default image
- User-created image

Setting Up the System Recovery Environment

To set up the system recovery environment, you need the following:

- A DA-820 computer
- A USB drive with minimum 4 GB of storage space
- A copy of the recovery suite

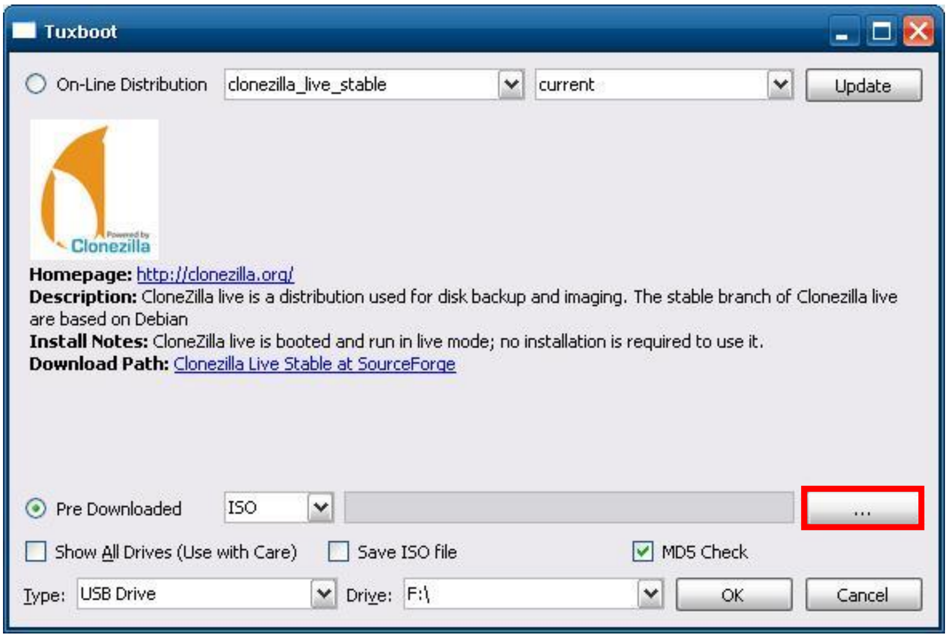
The following is an overview of the steps to setting up the system recovery environment. The subsequent sections include detailed information. If you have already created an image on a USB drive, skip to Step 3.

1. From the DA-820 software CD/DVD, copy the bootable recovery environment (an ISO image) to the USB drive.
2. Choose a recovery image type to create.
3. If you choose to create a bare-bones image and use this image for system recovery, any applications or scripts you install later will be lost if a recovery is required.
4. Configure the BIOS to have the system boot from the USB port first during startup.
5. When the system next restarts, it will boot into the Clonezilla recovery environment on the USB drive.
6. Create a copy of a fully configured system on the USB drive. This is the alternative to the stock OS recovery method in Step 2.
7. Perform a system recovery. You can also perform this step to test the setup.
8. Reset the BIOS back to its original settings.

Recovering from the Factory Default Image

Step 1: Preparing the USB Drive

1. In the **\recovery\DA-820-LX_Recovery\clonezilla** directory on the DA-820 software CD/DVD, start the Clonezilla imaging program (within the current OS) by running **tuxboot-windows-23.exe**.
2. In the Tuxboot screen, select **Pre-Downloaded** and select **ISO** from the drop-down list.
3. Click the ellipsis (**...**) button to select the Clonezilla ISO image on the software CD/DVD.



Tuxboot

○ On-Line Distribution clonezilla_live_stable current Update

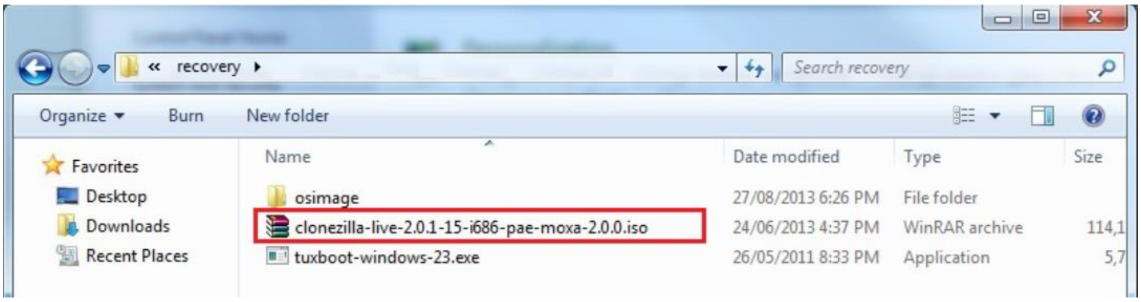
Homepage: http://clonezilla.org/
Description: CloneZilla live is a distribution used for disk backup and imaging. The stable branch of Clonezilla live are based on Debian
Install Notes: CloneZilla live is booted and run in live mode; no installation is required to use it.
Download Path: Clonezilla Live Stable at SourceForge

● Pre Downloaded ISO ...

□ Show All Drives (Use with Care) □ Save ISO file ☑ MD5 Check

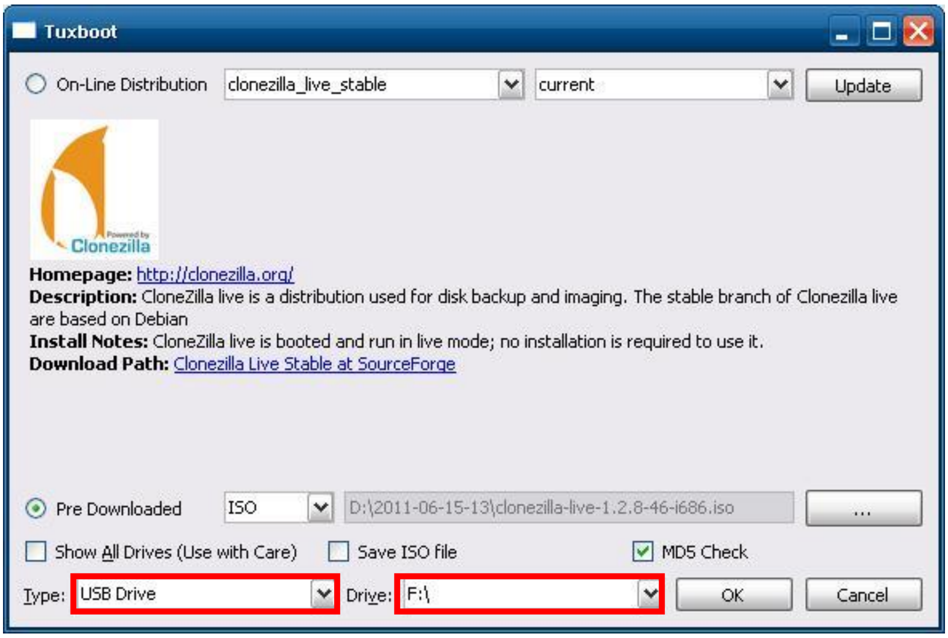
Type: USB Drive Drive: F:\ OK Cancel

4. In the **\Recovery\DA-820-LX_Recovery\clonezilla** directory on the software DVD, select the Clonezilla recovery environment ISO image.



Name	Date modified	Type	Size
osimage	27/08/2013 6:26 PM	File folder	
clonezilla-live-2.0.1-15-i686-pae-moxa-2.0.0.iso	24/06/2013 4:37 PM	WinRAR archive	114,1
tuxboot-windows-23.exe	26/05/2011 8:33 PM	Application	5,7

5. In the Tuxboot screen, configure the following fields:
   - **Type:** Select **USB Drive** from the drop-down list.
   - **Drive:** Select a letter for the drive on which the USB drive is mounted.



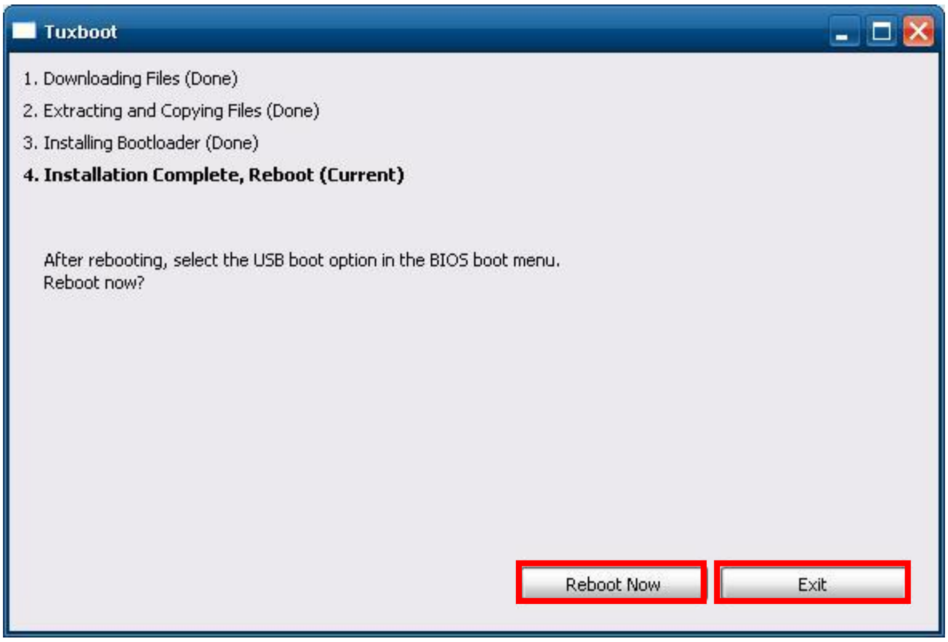
Homepage: http://clonezilla.org/
Description: CloneZilla live is a distribution used for disk backup and imaging. The stable branch of Clonezilla live are based on Debian
Install Notes: CloneZilla live is booted and run in live mode; no installation is required to use it.
Download Path: Clonezilla Live Stable at SourceForge

● Pre Downloaded ISO D:\2011-06-15-13\clonezilla-live-1.2.8-46-i686.iso ...

□ Show All Drives (Use with Care) □ Save ISO file ☑ MD5 Check

Type: USB Drive Drive: F:\ OK Cancel

6. Click **OK**. The system copies the Clonezilla recovery environment and the boot loader to the USB drive.



1. Downloading Files (Done)
2. Extracting and Copying Files (Done)
3. Installing Bootloader (Done)
4. **Installation Complete, Reboot (Current)**

After rebooting, select the USB boot option in the BIOS boot menu.
Reboot now?

7. Click **Exit** to close the application.

NOTE You must delete the EFI directory on the USB drive.

8. From the **/media/cd0/recovery/os_image** folder on the software CD/DVD, copy the operating system image to the **/media/usb0/home/partimag** folder on the USB drive.

   The following figure shows a command example to copy the image file.

```
moxa@Moxa:~# cp -a /media/cd0/recovery/os_image  /media/usb0/home/partimag/
```

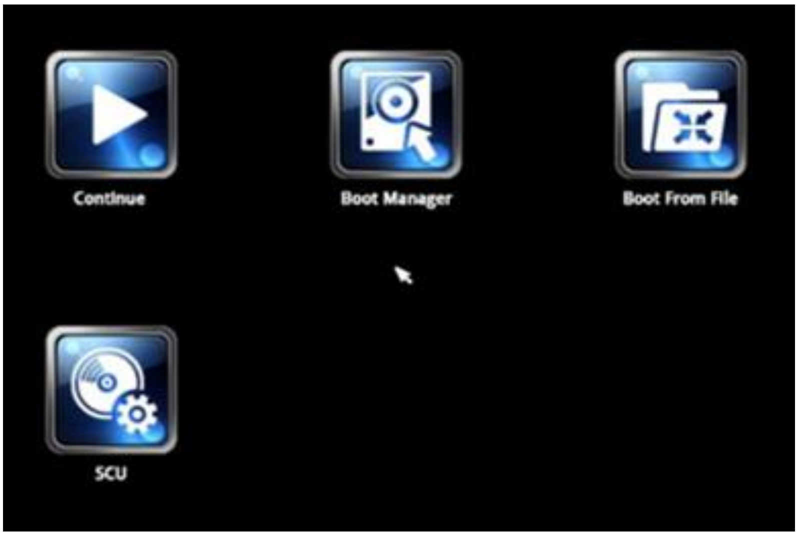
You have created a USB recovery drive that enables you to perform a system recovery on the DA-820 using the factory default image.

To create your own system recovery image, see the Creating a Custom System Image.

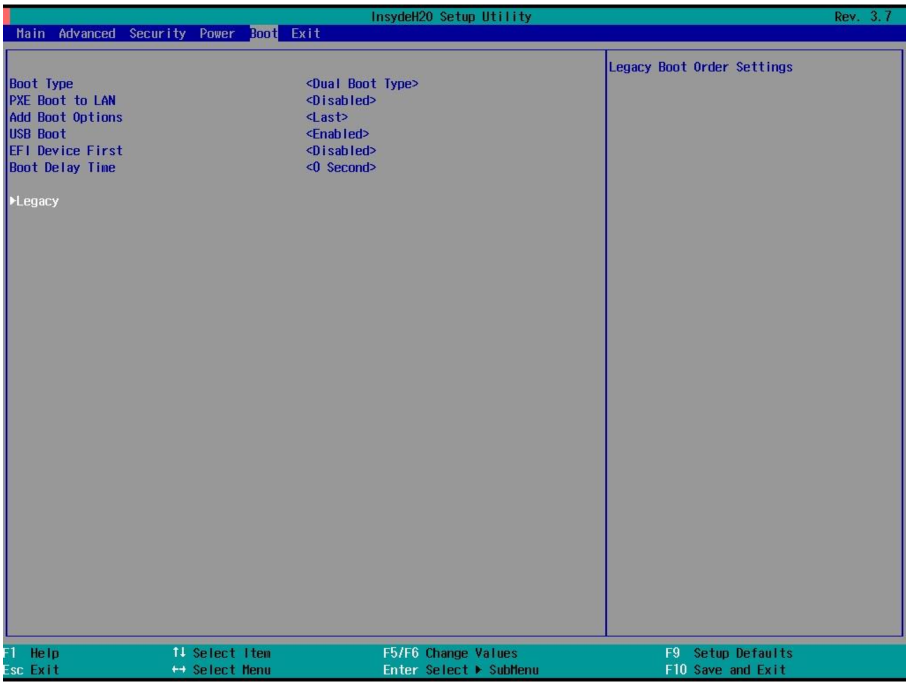
Step 2: Setting the BIOS to Boot from the USB Drive

This section shows you how to configure the BIOS to set the system to boot from a USB drive.

1. Restart the DA-820 and press **F2** during the POST process until you hear a long beep.
2. In the BIOS configuration screen, click **SCU** to enter the BIOS setup menu.

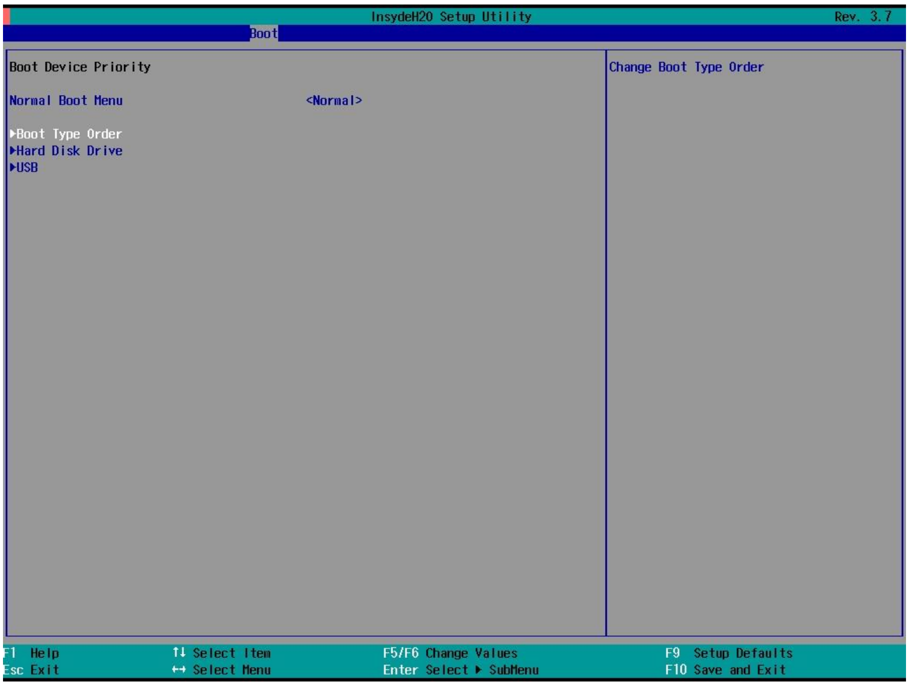


3. Use the left or right arrow key to navigate to the **Boot** tab; then, press [Enter].
4. In the **Boot** screen, use the up or down arrow key to select **Legacy** and press [Enter].



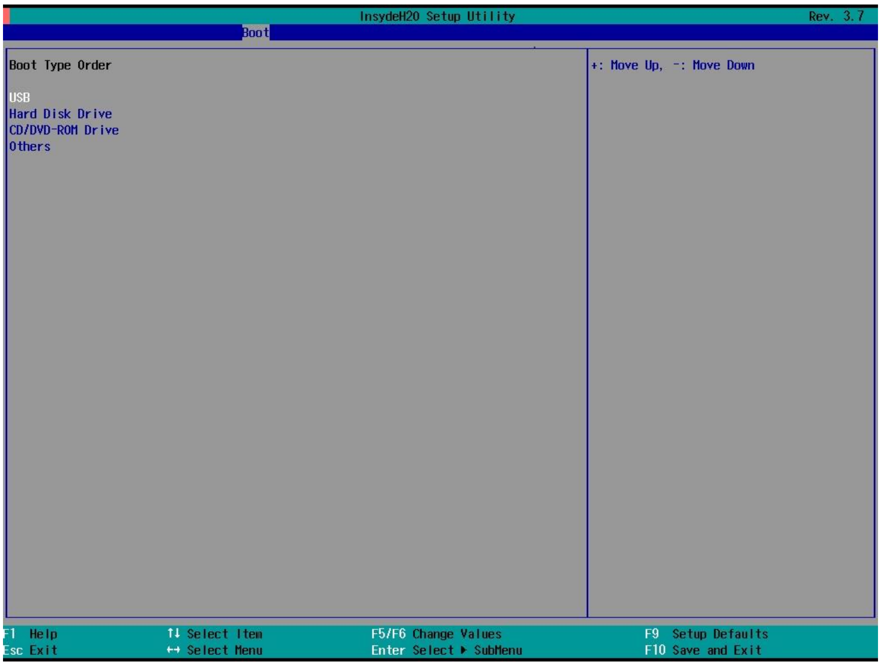
```
                                  InsydeH20 Setup Utility                                Rev. 3.7
 Main  Advanced  Security  Power  Boot  Exit

                                                                  Legacy Boot Order Settings
Boot Type                                  <Dual Boot Type>
PXE Boot to LAN                            <Disabled>
Add Boot Options                           <Last>
USB Boot                                   <Enabled>
EFI Device First                           <Disabled>
Boot Delay Time                            <0 Second>

▸Legacy

F1  Help            ↑↓ Select Item     F5/F6 Change Values           F9  Setup Defaults
Esc Exit            ↔ Select Menu      Enter Select ▸ SubMenu        F10 Save and Exit
```

5. Use the up or down arrow key to select **Boot Type Order** and press [Enter].



```
InsydeH20 Setup Utility                                   Rev. 3.7
                              Boot

Boot Device Priority                                Change Boot Type Order

Normal Boot Menu                <Normal>

▶Boot Type Order
▶Hard Disk Drive
▶USB

F1  Help        ↑↓ Select Item    F5/F6 Change Values       F9  Setup Defaults
Esc Exit        ↔ Select Menu     Enter Select ▶ SubMenu    F10 Save and Exit
```

6. Use the up or down arrow key to select **USB** and use the plus or minus signs (**+ -**) to move the option to the first boot priority position.



```
InsydeH20 Setup Utility                                   Rev. 3.7
                              Boot

Boot Type Order                                     +: Move Up,  -: Move Down

USB
Hard Disk Drive
CD/DVD-ROM Drive
Others

F1  Help        ↑↓ Select Item    F5/F6 Change Values       F9  Setup Defaults
Esc Exit        ↔ Select Menu     Enter Select ▶ SubMenu    F10 Save and Exit
```

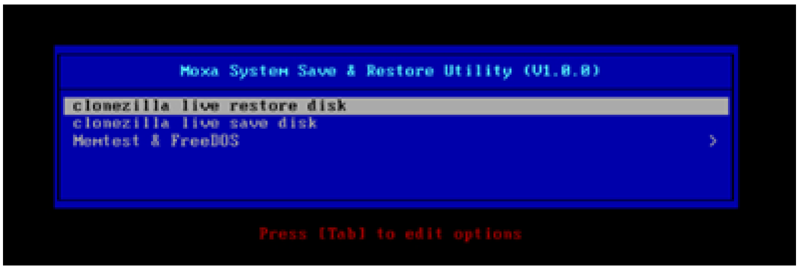
Step 3: Performing a System Recovery

Connect the USB drive to any of the DA-820's USB ports and then reboot the computer. The system will boot from the USB into the Clonezilla boot loader.

1. Connect the USB drive to a USB port on the DA-820 and restart the DA-820.

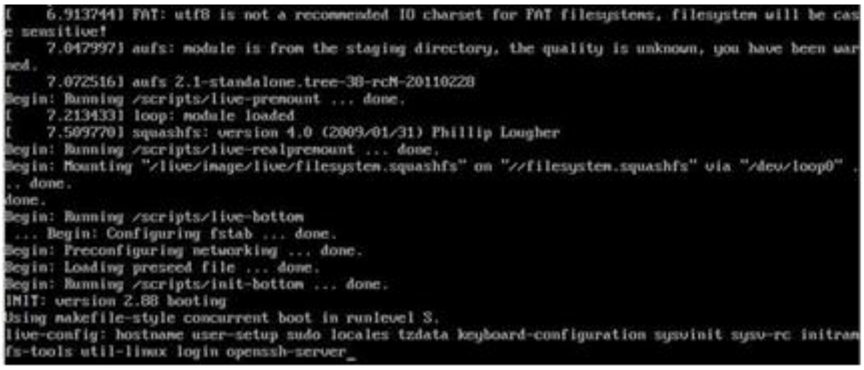
   The system boots from the USB drive.

2. Select **Clonezilla Live Restore Disk** to boot into the system restoration environment.



```
Moxa System Save & Restore Utility (V1.0.0)

clonezilla live restore disk
clonezilla live save disk
Memtest & FreeDOS                                    >

Press [Tab] to edit options
```

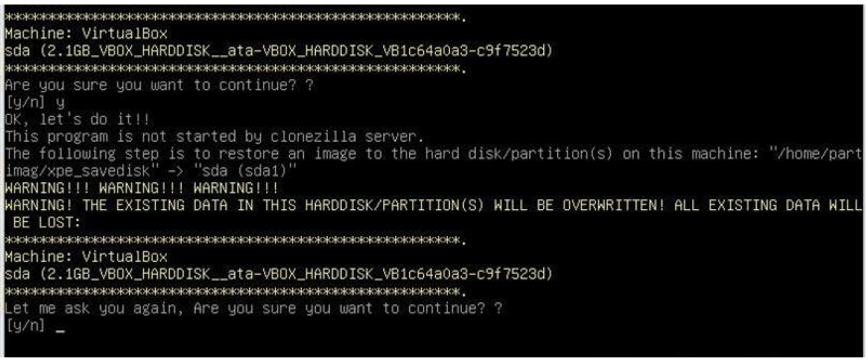
   Wait for the boot process to complete.



```
[    6.913744] FAT: utf8 is not a recommended IO charset for FAT filesystems, filesystem will be cas
e sensitive!
[    7.047997] aufs: module is from the staging directory, the quality is unknown, you have been war
ned.
[    7.072516] aufs 2.1-standalone.tree-38-rcN-20110228
Begin: Running /scripts/live-premount ... done.
[    7.213433] loop: module loaded
[    7.509770] squashfs: version 4.0 (2009/01/31) Phillip Lougher
Begin: Running /scripts/live-realpremount ... done.
Begin: Mounting "/live/image/live/filesystem.squashfs" on "//filesystem.squashfs" via "/dev/loop0" .
.. done.
done.
Begin: Running /scripts/live-bottom
 ... Begin: Configuring fstab ... done.
Begin: Preconfiguring networking ... done.
Begin: Loading preseed file ... done.
Begin: Running /scripts/init-bottom ... done.
INIT: version 2.88 booting
Using makefile-style concurrent boot in runlevel S.
live-config: hostname user-setup sudo locales tzdata keyboard-configuration sysvinit sysv-rc initram
fs-tools util-linux login openssh-server_
```

NOTE To cancel the system recovery process and exit from the Clonezilla boot loader, press any letter or press Ctrl+C.

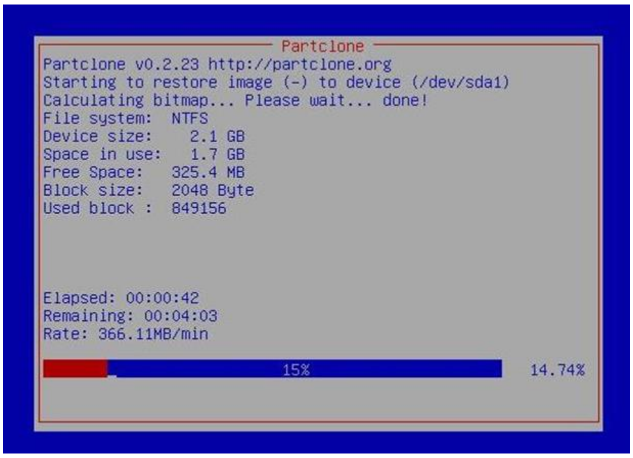
3. A message appears warning you that you are about to overwrite the hard drive and erase all data on the partition listed (for example, sda1). Enter **Y** (case insensitive).



```
*****************************************************.
Machine: VirtualBox
sda (2.1GB_VBOX_HARDDISK__ata-VBOX_HARDDISK_VB1c64a0a3-c9f7523d)
*****************************************************.
Are you sure you want to continue? ?
[y/n] y
OK, let's do it!!
This program is not started by clonezilla server.
The following step is to restore an image to the hard disk/partition(s) on this machine: "/home/part
imag/xpe_savedisk" -> "sda (sda1)"
WARNING!!! WARNING!!! WARNING!!!
WARNING! THE EXISTING DATA IN THIS HARDDISK/PARTITION(S) WILL BE OVERWRITTEN! ALL EXISTING DATA WILL
 BE LOST:
*****************************************************.
Machine: VirtualBox
sda (2.1GB_VBOX_HARDDISK__ata-VBOX_HARDDISK_VB1c64a0a3-c9f7523d)
*****************************************************.
Let me ask you again, Are you sure you want to continue? ?
[y/n] _
```

The Clonezilla boot loader copies the system image from the USB drive to the primary system drive. This erases all existing data on the primary system drive.

Wait for the process to complete; depending on the system, this might take up to 10 minutes.

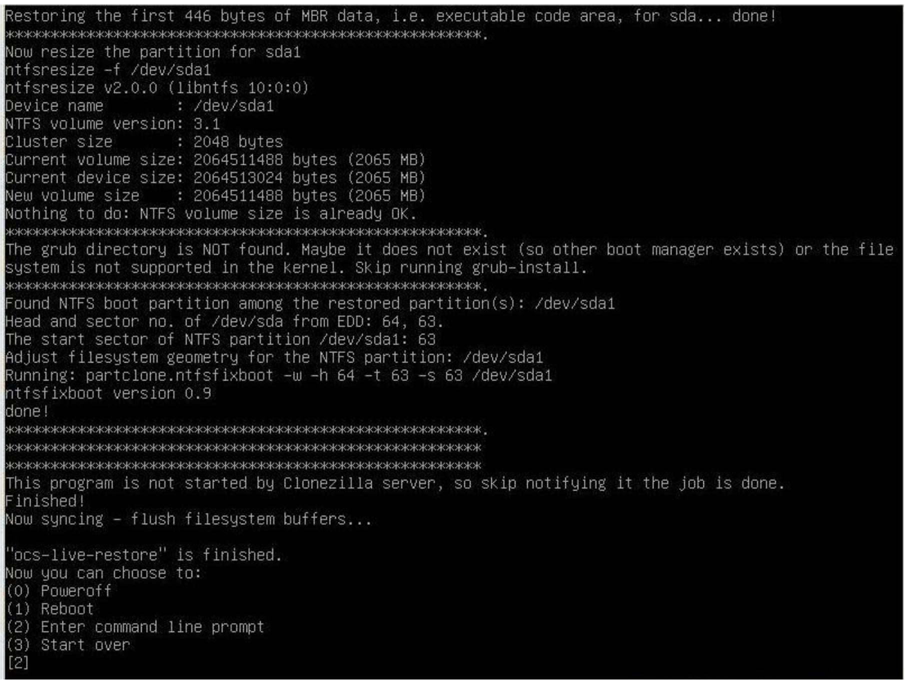


```
Partclone
Partclone v0.2.23 http://partclone.org
Starting to restore image (-) to device (/dev/sda1)
Calculating bitmap... Please wait... done!
File system:  NTFS
Device size:     2.1 GB
Space in use:    1.7 GB
Free Space:    325.4 MB
Block size:    2048 Byte
Used block :   849156



Elapsed: 00:00:42
Remaining: 00:04:03
Rate: 366.11MB/min

                  15%                    14.74%
```

4. Complete the restoration process by selecting (**0**) **Poweroff** to shut down the computer.

WARNING

If the Power Switch remains inserted on the front panel of the computer and is at the ON position, the system performs a soft reboot. To prevent a soft reboot, either use the switch to turn off the computer immediately following the shutdown or may remove the power switch from the front panel and then use the console to shut down the computer by pressing 0.



```
Restoring the first 446 bytes of MBR data, i.e. executable code area, for sda... done!
*****************************************************.
Now resize the partition for sda1
ntfsresize -f /dev/sda1
ntfsresize v2.0.0 (libntfs 10:0:0)
Device name        : /dev/sda1
NTFS volume version: 3.1
Cluster size       : 2048 bytes
Current volume size: 2064511488 bytes (2065 MB)
Current device size: 2064513024 bytes (2065 MB)
New volume size    : 2064511488 bytes (2065 MB)
Nothing to do: NTFS volume size is already OK.
*****************************************************.
The grub directory is NOT found. Maybe it does not exist (so other boot manager exists) or the file
system is not supported in the kernel. Skip running grub-install.
*****************************************************.
Found NTFS boot partition among the restored partition(s): /dev/sda1
Head and sector no. of /dev/sda from EDD: 64, 63.
The start sector of NTFS partition /dev/sda1: 63
Adjust filesystem geometry for the NTFS partition: /dev/sda1
Running: partclone.ntfsfixboot -w -h 64 -t 63 -s 63 /dev/sda1
ntfsfixboot version 0.9
done!
*****************************************************.
*****************************************************
*****************************************************
This program is not started by Clonezilla server, so skip notifying it the job is done.
Finished!
Now syncing - flush filesystem buffers...

"ocs-live-restore" is finished.
Now you can choose to:
(0) Poweroff
(1) Reboot
(2) Enter command line prompt
(3) Start over
[2]
```

5. After the computer is turned off, remove the USB drive and store it in a safe location.

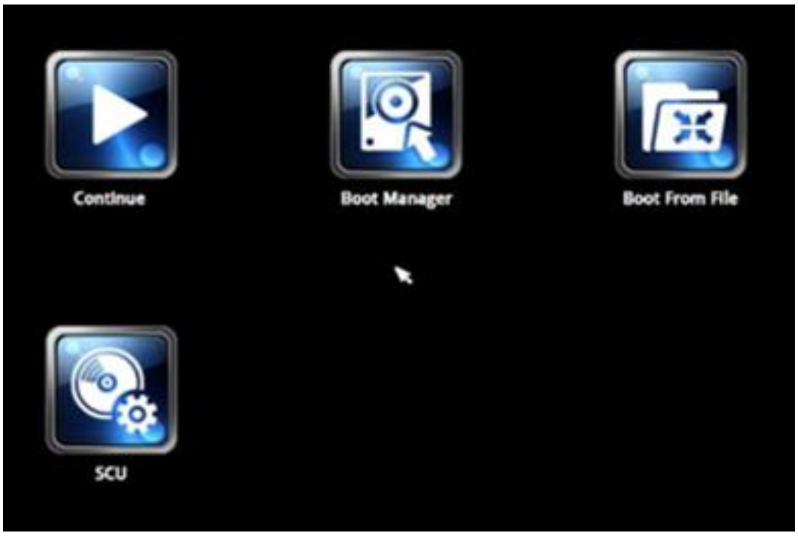
Step 4: Resetting the BIOS

After you have performed a system recovery, reset the BIOS to enable the system to book from the disk drive. This prevents the system from booting from an unauthorized USB drives and provides added system security during startup.

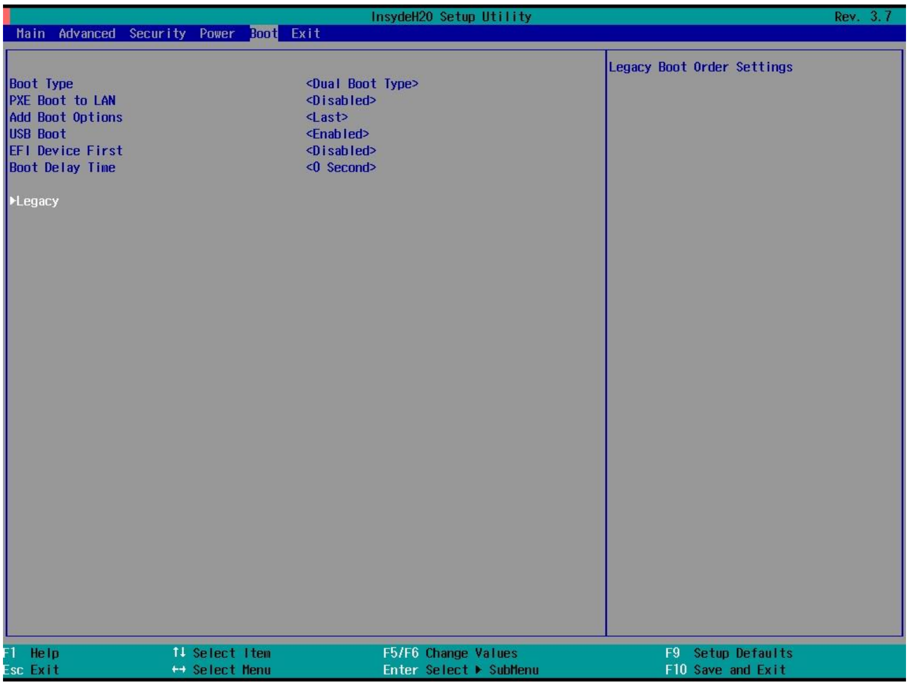
ATTENTION

The Da-820 does not differentiate a bootable and non-bootable USB drive. If you set the DA-820 to boot from a USB drive, the startup process will terminate if a non-bootable USB drive is connect to a USB port on the DA-820.

1. Restart the DA-820 and press **F2** during the POST process until you hear a long beep.
2. In the BIOS configuration screen, click **SCU** to enter the BIOS setup menu.

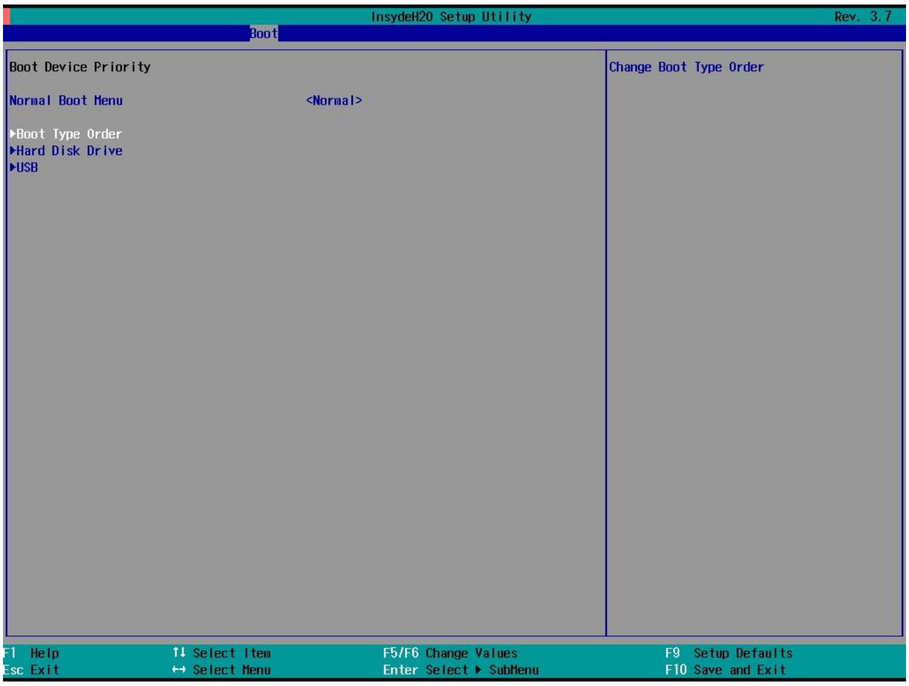


3. Use the left or right arrow key to navigate to the **Boot** tab; then, press [Enter].
4. In the **Boot** screen, use the up or down arrow key to select **Legacy** and press [Enter].



```
                              InsydeH20 Setup Utility                      Rev. 3.7
 Main  Advanced  Security  Power  Boot  Exit

                                                             Legacy Boot Order Settings
Boot Type                           <Dual Boot Type>
PXE Boot to LAN                     <Disabled>
Add Boot Options                    <Last>
USB Boot                            <Enabled>
EFI Device First                    <Disabled>
Boot Delay Time                     <0 Second>

▸Legacy

F1  Help          ↑↓ Select Item       F5/F6 Change Values          F9  Setup Defaults
Esc Exit          ↔ Select Menu        Enter Select ▸ SubMenu       F10 Save and Exit
```

5. Use the up or down arrow keys to select **Boot Type Order** and press [Enter].



InsydeH20 Setup Utility Rev. 3.7

Boot

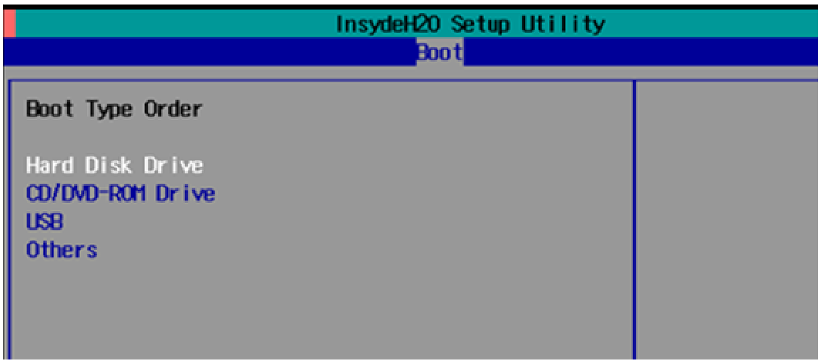
Boot Device Priority

Normal Boot Menu <Normal>

▸Boot Type Order
▸Hard Disk Drive
▸USB

Change Boot Type Order

6. Use the up or down arrow key to select **Hard Disk Drive** and use the plus or minus signs (**+ -**) to move the option to the first boot priority position.



InsydeH20 Setup Utility

Boot

Boot Type Order

Hard Disk Drive
CD/DVD-ROM Drive
USB
Others

7. Press **F10** and press [Enter] to save the changes and exit from the BIOS configuration interface.

   The system automatically reboots from the hard disk drive.

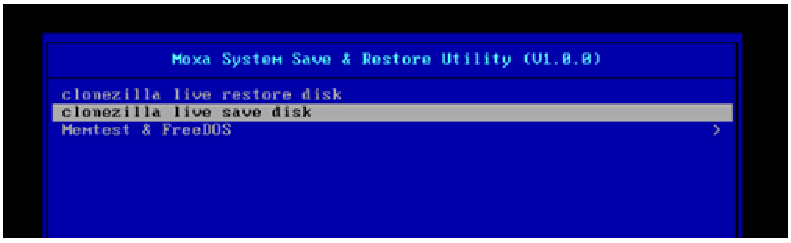
Creating a Custom System Image

This section shows you how to create a system recovery image from a system with customized applications and scripts.

The procedure below describes a configuration for restoring a complete system that has been customized with user applications and scripts. Here, you will save to the USB drive a copy of the entire system as it is currently configured to be used as a full system recovery image should the system crash. During this process, ***all files on your USB that are mounted under*** `F:\home\partimag\` ***will be overwritten***.

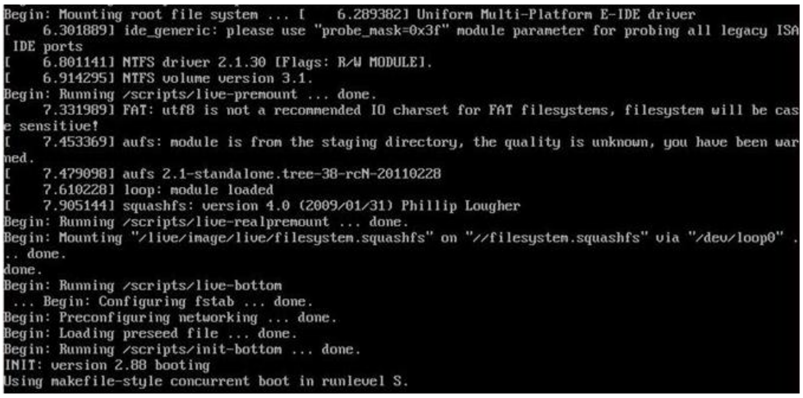
1. gjfghj
2. Configure the BIOS to set the system to boot from a USB drive. For more information, see **Step 2: Setting the BIOS to Boot from the USB Drive**.
3. Restart the DA-820.

4. In the System Save & Recovery Utility screen, select **clonezilla live save disk** and press [Enter].



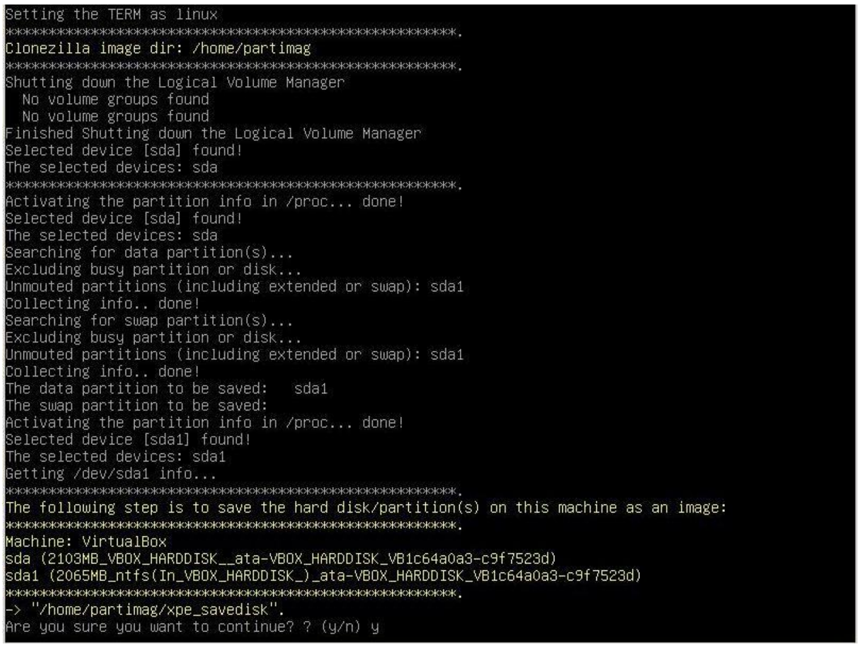
```
Moxa System Save & Restore Utility (V1.0.0)

clonezilla live restore disk
clonezilla live save disk
Memtest & FreeDOS                                              >
```

The DA-820 boots into the image creation environment. Wait for the boot process to complete.



```
Begin: Mounting root file system ... [    6.289382] Uniform Multi-Platform E-IDE driver
[    6.301889] ide_generic: please use "probe_mask=0x3f" module parameter for probing all legacy ISA
 IDE ports
[    6.801141] NTFS driver 2.1.30 [Flags: R/W MODULE].
[    6.914295] NTFS volume version 3.1.
Begin: Running /scripts/live-premount ... done.
[    7.331989] FAT: utf8 is not a recommended IO charset for FAT filesystems, filesystem will be cas
e sensitive!
[    7.453369] aufs: module is from the staging directory, the quality is unknown, you have been war
ned.
[    7.479098] aufs 2.1-standalone.tree-38-rcN-20110228
[    7.610228] loop: module loaded
[    7.905144] squashfs: version 4.0 (2009/01/31) Phillip Lougher
Begin: Running /scripts/live-realpremount ... done.
Begin: Mounting "/live/image/live/filesystem.squashfs" on "//filesystem.squashfs" via "/dev/loop0" .
.. done.
done.
Begin: Running /scripts/live-bottom
 ... Begin: Configuring fstab ... done.
Begin: Preconfiguring networking ... done.
Begin: Loading preseed file ... done.
Begin: Running /scripts/init-bottom ... done.
INIT: version 2.88 booting
Using makefile-style concurrent boot in runlevel S.
```

5. A message appears warning you that the system is about to overwrite the **/home/partimag** folder on the USB drive and erase all data in the folder. Enter **Y** (case insensitive).



```
Setting the TERM as linux
*****************************************************.
Clonezilla image dir: /home/partimag
*****************************************************.
Shutting down the Logical Volume Manager
  No volume groups found
  No volume groups found
Finished Shutting down the Logical Volume Manager
Selected device [sda] found!
The selected devices: sda
*****************************************************.
Activating the partition info in /proc... done!
Selected device [sda] found!
The selected devices: sda
Searching for data partition(s)...
Excluding busy partition or disk...
Unmouted partitions (including extended or swap): sda1
Collecting info.. done!
Searching for swap partition(s)...
Excluding busy partition or disk...
Unmouted partitions (including extended or swap): sda1
Collecting info.. done!
The data partition to be saved:   sda1
The swap partition to be saved:
Activating the partition info in /proc... done!
Selected device [sda1] found!
The selected devices: sda1
Getting /dev/sda1 info...
*****************************************************.
The following step is to save the hard disk/partition(s) on this machine as an image:
*****************************************************.
Machine: VirtualBox
sda (2103MB_VBOX_HARDDISK__ata-VBOX_HARDDISK_VB1c64a0a3-c9f7523d)
sda1 (2065MB_ntfs(In_VBOX_HARDDISK_)_ata-VBOX_HARDDISK_VB1c64a0a3-c9f7523d)
*****************************************************.
-> "/home/partimag/xpe_savedisk".
Are you sure you want to continue? ? (y/n) y
```

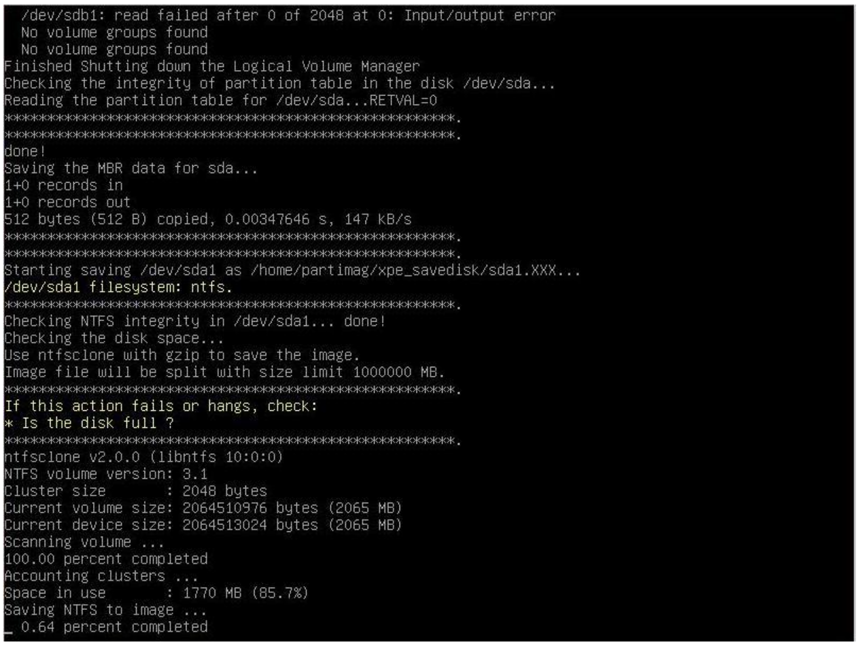
WARNING

Regardless of system image type (factory default or custom image), the system creates a recovery image using the same filename. This means that you cannot save more than one system image on a USB drive.

The system copies all data on the hard drive to the USB drive. This might take up to half an hour, depending on the amount of data to copy. Wait until the process is complete.

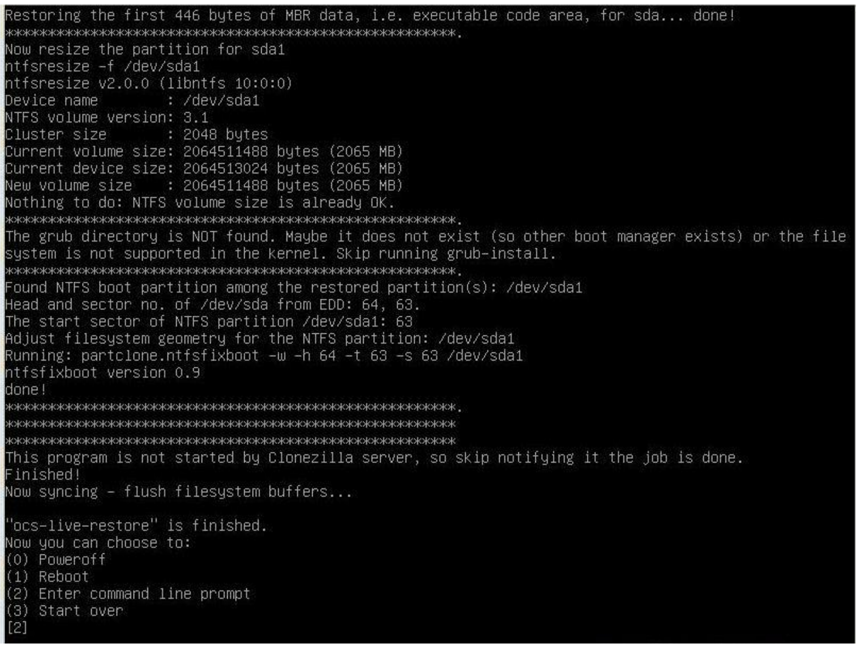
ATTENTION

Do NOT remove the USB drive from the DA-820 during the image creation process.



```
  /dev/sdb1: read failed after 0 of 2048 at 0: Input/output error
  No volume groups found
  No volume groups found
Finished Shutting down the Logical Volume Manager
Checking the integrity of partition table in the disk /dev/sda...
Reading the partition table for /dev/sda...RETVAL=0
*****************************************************.
*****************************************************.
done!
Saving the MBR data for sda...
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.00347646 s, 147 kB/s
*****************************************************.
*****************************************************.
Starting saving /dev/sda1 as /home/partimag/xpe_savedisk/sda1.XXX...
/dev/sda1 filesystem: ntfs.
*****************************************************.
Checking NTFS integrity in /dev/sda1... done!
Checking the disk space...
Use ntfsclone with gzip to save the image.
Image file will be split with size limit 1000000 MB.
*****************************************************.
If this action fails or hangs, check:
* Is the disk full ?
*****************************************************.
ntfsclone v2.0.0 (libntfs 10:0:0)
NTFS volume version: 3.1
Cluster size       : 2048 bytes
Current volume size: 2064510976 bytes (2065 MB)
Current device size: 2064513024 bytes (2065 MB)
Scanning volume ...
100.00 percent completed
Accounting clusters ...
Space in use       : 1770 MB (85.7%)
Saving NTFS to image ...
_ 0.64 percent completed
```

6. Turn off the DA-820. Type **0** and press [Enter].



```
Restoring the first 446 bytes of MBR data, i.e. executable code area, for sda... done!
*****************************************************.
Now resize the partition for sda1
ntfsresize -f /dev/sda1
ntfsresize v2.0.0 (libntfs 10:0:0)
Device name        : /dev/sda1
NTFS volume version: 3.1
Cluster size       : 2048 bytes
Current volume size: 2064511488 bytes (2065 MB)
Current device size: 2064513024 bytes (2065 MB)
New volume size    : 2064511488 bytes (2065 MB)
Nothing to do: NTFS volume size is already OK.
*****************************************************.
The grub directory is NOT found. Maybe it does not exist (so other boot manager exists) or the file
system is not supported in the kernel. Skip running grub-install.
*****************************************************.
Found NTFS boot partition among the restored partition(s): /dev/sda1
Head and sector no. of /dev/sda from EDD: 64, 63.
The start sector of NTFS partition /dev/sda1: 63
Adjust filesystem geometry for the NTFS partition: /dev/sda1
Running: partclone.ntfsfixboot -w -h 64 -t 63 -s 63 /dev/sda1
ntfsfixboot version 0.9
done!
*****************************************************.
*****************************************************
*****************************************************
This program is not started by Clonezilla server, so skip notifying it the job is done.
Finished!
Now syncing - flush filesystem buffers...

"ocs-live-restore" is finished.
Now you can choose to:
(0) Poweroff
(1) Reboot
(2) Enter command line prompt
(3) Start over
[2]
```

7. After the DA-820 is turned off, remove the USB drive and store it in a safe location.
8. Reset the BIOS to set the system to boot from the hard disk drive. You may test the image by performing a system recovery.

A

Software Components

Package Name	Version	Description
acpi	1.6-1	amd64 displays information on ACPI devices
acpi-support-base	0.140-5+deb7u2	Contains scripts for handling base ACPI events (for example, the power button)
acpid	1:2.0.16-1+deb7u1	amd64 Advanced Configuration and Power Interface event daemon
adduser	3.113+nmu3	To add and remove users and groups
alsa-base	1.0.25+3~deb7u1	ALSA driver configuration files
alsa-utils	1.0.25-4	amd64 Utilities for configuring and using ALSA
apache2	2.2.22-13+deb7u3	amd64 Apache HTTP Server metapackage
apache2-mpm-prefork	2.2.22-13+deb7u3	amd64 Apache HTTP Server - traditional non-threaded model
apache2-utils	2.2.22-13+deb7u3	amd64 utility programs for webservers
apache2.2-bin	2.2.22-13+deb7u3	amd64 Apache HTTP Server common binary files
apache2.2-common	2.2.22-13+deb7u3	amd64 Apache HTTP Server common files
apt	0.9.7.9+deb7u2	amd64 commandline package manager
apt-utils	0.9.7.9+deb7u2	amd64 package managment related utility programs
aptitude	0.6.8.2-1	amd64 terminal-based package manager
aptitude-common	0.6.8.2-1	All architecture indepedent files for the aptitude package manager
base-files	7.1wheezy6	amd64 Debian base system miscellaneous files
base-passwd	3.5.26	amd64 Debian base system master password and group files
bash	4.2+dfsg-0.1	amd64 GNU Bourne Again SHell
bash-completion	1:2.0-1	All programmable completion for the bash shell
binutils	2.22-8	amd64 GNU assembler, linker and binary utilities
bridge-utils	1.5-6	amd64 utilities for configuring the Linux Ethernet bridge
bsdmainutils	9.0.3	amd64 collection of more utilities from FreeBSD
bsdutils	1:2.20.1-5.3	amd64 Basic utilities from 4.4BSD-Lite
build-essential	11.5	amd64 informational list of build-essential packages
busybox	1:1.20.0-7	amd64 tiny utilities for small and embedded systems
bzip2	1.0.6-4	amd64 high-quality block-sorting file compressor – utilities

Package Name	Version	Description
console-setup	1.88	All console font and keymap setup program
console-setup-linux	1.88	All Linux specific part of console-setup
coreutils	8.13-3.5	amd64 GNU core utilities
cpio	2.11+dfsg-0.1	amd64 GNU cpio (a program to manage archives of files)
cpp	4:4.7.2-1	amd64 GNU C preprocessor (cpp)
cpp-4.6	4.6.3-14	amd64 GNU C preprocessor
cpp-4.7	4.7.2-5	amd64 GNU C preprocessor
cron	3.0pl1-124	amd64 process scheduling daemon
da820-irigb	1.0	amd64 Moxa IRIG-B time sync daemon
dash	0.5.7-3	amd64 POSIX-compliant shell
debconf	1.5.49	All Debian configuration management system
debconf-i18n	1.5.49	All full internationalization support for debconf
debian-archive-keyring	2012.4	All GnuPG archive keys of the Debian archive
debianutils	4.3.2	amd64 Miscellaneous utilities specific to Debian
dialog	1.1-20120215-2	amd64 Displays user-friendly dialog boxes from shell scripts
diffutils	1:3.2-6	amd64 File comparison utilities
discover	2.1.2-5.2	amd64 hardware identification system
discover-data	2.2010.10.18	All data lists for Discover hardware detection system
dmidecode	2.11-9	amd64 SMBIOS/DMI table decoder
dmsetup	2:1.02.74-8	amd64 Linux Kernel Device Mapper userspace library
dpkg	1.16.15	amd64 Debian package management system
dpkg-dev	1.16.15	All Debian package development tools
e2fslibs:amd64	1.42.5-1.1	amd64 ext2/ext3/ext4 file system libraries
e2fsprogs	1.42.5-1.1	amd64 ext2/ext3/ext4 file system utilities
ethtool	1:3.4.2-1	amd64 display or change Ethernet device settings
exim4-base	4.80-7	amd64 support files for all Exim MTA (v4) packages
exim4-config	4.80-7	All configuration for the Exim MTA (v4)
exim4-daemon-light	4.80-7	amd64 lightweight Exim MTA (v4) daemon
fakeroot	1.18.4-2	amd64 tool for simulating superuser privileges
findutils	4.4.2-4	amd64 utilities for finding files--find, xargs
firmware-linux-free	3.2	All Binary firmware for various drivers in the Linux kernel
g++	4:4.7.2-1	amd64 GNU C++ compiler
g++-4.7	4.7.2-5	amd64 GNU C++ compiler
gcc	4:4.7.2-1	amd64 GNU C compiler
gcc-4.6	4.6.3-14	amd64 GNU C compiler
gcc-4.6-base:amd64	4.6.3-14	amd64 GCC, the GNU Compiler Collection (base package)
gcc-4.7	4.7.2-5	amd64 GNU C compiler
gcc-4.7-base:amd64	4.7.2-5	amd64 GCC, the GNU Compiler Collection (base package)
gettext-base	0.18.1.1-9	amd64 GNU Internationalization utilities for the base system

Package Name	Version	Description
gnupg	1.4.12-7+deb7u4	amd64 GNU privacy guard - a free PGP replacement
gpgv	1.4.12-7+deb7u4	amd64 GNU privacy guard - signature verification tool
grep	2.12-2	amd64 GNU grep, egrep and fgrep
groff-base	1.21-9	amd64 GNU troff text-formatting system (base system components)
grub-common	1.99-27+deb7u2	amd64 GRand Unified Bootloader (common files)
grub-pc	1.99-27+deb7u2	amd64 GRand Unified Bootloader, version 2 (PC/BIOS version)
grub-pc-bin	1.99-27+deb7u2	amd64 GRand Unified Bootloader, version 2 (PC/BIOS binaries)
grub2-common	1.99-27+deb7u2	amd64 GRand Unified Bootloader (common files for version 2)
gzip	1.5-1.1	amd64 GNU compression utilities
hdparm	9.39-1+b1	amd64 tune hard disk parameters for high performance
heirloom-mailx	12.5-2	amd64 feature-rich BSD mail(1)
hostname	3.11	amd64 utility to set/show the host name or domain name
ifupdown	0.7.8	amd64 high level tools to configure network interfaces
initramfs-tools	0.115~bpo70+1	All generic modular initramfs generator
initscripts	2.88dsf-41+deb7u1	amd64 scripts for initializing and shutting down the system
insserv	1.14.0-5	amd64 boot sequence organizer using LSB init.d script dependency information
install-info	4.13a.dfsg.1-10	amd64 Manage installed documentation in info format
iproute	20120521-3+b3	amd64 networking and traffic control tools
iptables	1.4.14-3.1	amd64 administration tools for packet filtering and NAT
iputils-ping	3:20101006-1+b1	amd64 Tools to test the reachability of network hosts
isc-dhcp-client	4.2.2.dfsg.1-5+deb70u6	amd64 ISC DHCP client
isc-dhcp-common	4.2.2.dfsg.1-5+deb70u6	amd64 common files used by all the isc-dhcp* packages
kbd	1.15.3-9	amd64 Linux console font and keytable utilities
keyboard-configuration	1.88	All system-wide keyboard preferences
klibc-utils	2.0.1-3.1	amd64 small utilities built with klibc for early boot
kmod	9-3	amd64 tools for managing Linux kernel modules
laptop-detect	0.13.7	amd64 attempt to detect a laptop
libacl1:amd64	2.2.51-8	amd64 Access control list shared library
libalgorithm-diff-perl	1.19.02-2	All modules to find differences between files
libalgorithm-diff-xs-perl	0.04-2+b1	amd64 module to find differences between files (XS accelerated)
libalgorithm-merge-perl	0.08-2	All Perl modules for three-way merge of textual data

Package Name	Version	Description
libapache2-mod-php5	5.4.4-14+deb7u12	amd64 server-side, HTML-embedded scripting language (Apache 2 module)
libapr1	1.4.6-3+deb7u1	amd64 Apache Portable Runtime Library
libaprutil1	1.4.1-3	amd64 Apache Portable Runtime Utility Library
libaprutil1-dbd-sqlite3	1.4.1-3	amd64 Apache Portable Runtime Utility Library - SQLite3 Driver
libaprutil1-ldap	1.4.1-3	amd64 Apache Portable Runtime Utility Library - LDAP Driver
libapt-inst1.5:amd64	0.9.7.9+deb7u2	amd64 deb package format runtime library
libapt-pkg4.12:amd64	0.9.7.9+deb7u2	amd64 package managment runtime library
libasound2:amd64	1.0.25-4	amd64 shared library for ALSA applications
libasprintf0c2:amd64	0.18.1.1-9	amd64 GNU library to use fprintf and friends in C++
libattr1:amd64	1:2.4.46-8	amd64 Extended attribute shared library
libblkid1:amd64	2.20.1-5.3	amd64 block device id library
libboost-iostreams1.49.0	1.49.0-3.2	amd64 Boost.Iostreams Library
libbsd0:amd64	0.4.2-1	amd64 utility functions from BSD systems - shared library
libbz2-1.0:amd64	1.0.6-4	amd64 high-quality block-sorting file compressor library - runtime
libc-bin	2.13-38+deb7u4	amd64 Embedded GNU C Library: Binaries
libc-dev-bin	2.13-38+deb7u4	amd64 Embedded GNU C Library: Development binaries
libc6:amd64	2.13-38+deb7u4	amd64 Embedded GNU C Library: Shared libraries
libc6-dev:amd64	2.13-38+deb7u4	amd64 Embedded GNU C Library: Development Libraries and Header Files
libcap2:amd64	1:2.22-1.2	amd64 support for getting/setting POSIX.1e capabilities
libclass-isa-perl	0.36-3	All reports the search path for a class's ISA tree
libcomerr2:amd64	1.42.5-1.1	amd64 common error description library
libcwidget3	0.5.16-3.4	amd64 high-level terminal interface library for C++ (runtime files)
libdb5.1:amd64	5.1.29-5	amd64 Berkeley v5.1 Database Libraries [runtime]
libdevmapper1.02.1:amd64	2:1.02.74-8	amd64 Linux Kernel Device Mapper userspace library
libdiscover2	2.1.2-5.2	amd64 hardware identification library
libdmraid1.0.0.rc16	1.0.0.rc16-4.2	amd64 Device-Mapper Software RAID support tool - shared library
libdpkg-perl	1.16.15	all Dpkg perl modules
libedit2:amd64	2.11-20080614-5	amd64 BSD editline and history libraries
libept1.4.12	1.0.9	amd64 High-level library for managing Debian package information
libexpat1:amd64	2.1.0-1+deb7u1	amd64 XML parsing C library - runtime library
libfile-copy-recursive-perl	0.38-1	All Perl extensions for recursively copying files and directories
libfile-fcntllock-perl	0.14-2	amd64 Perl module for file locking with fcntl(2)

Package Name	Version	Description
libfreetype6:amd64	2.4.9-1.1	amd64 FreeType 2 font engine, shared library files
libfuse2:amd64	2.9.0-2+deb7u1	amd64 Filesystem in Userspace (library)
libgcc1:amd64	1:4.7.2-5	amd64 GCC support library
libgcrypt11:amd64	1.5.0-5+deb7u1	amd64 LGPL Crypto library - runtime library
libgdbm3:amd64	1.8.3-11	amd64 GNU dbm database routines (runtime version)
libgmp10:amd64	2:5.0.5+dfsg-2	amd64 Multiprecision arithmetic library
libgnutls26:amd64	2.12.20-8+deb7u2	amd64 GNU TLS library - runtime library
libgomp1:amd64	4.7.2-5	amd64 GCC OpenMP (GOMP) support library
libgpg-error0:amd64	1.10-3.1	amd64 library for common error values and messages in GnuPG components
libgpm2:amd64	1.20.4-6	amd64 General Purpose Mouse - shared library
libgssapi-krb5-2:amd64	1.10.1+dfsg-5+deb7u2	amd64 MIT Kerberos runtime libraries - krb5 GSS-API Mechanism
libidn11:amd64	1.25-2	amd64 GNU Libidn library, implementation of IETF IDN specifications
libitm1:amd64	4.7.2-5	amd64 GNU Transactional Memory Library
libk5crypto3:amd64	1.10.1+dfsg-5+deb7u2	amd64 MIT Kerberos runtime libraries - Crypto Library
libkeyutils1:amd64	1.5.5-3	amd64 Linux Key Management Utilities (library)
libklibc	2.0.1-3.1	amd64 minimal libc subset for use with initramfs
libkmod2:amd64	9-3	amd64 libkmod shared library
libkrb5-3:amd64	1.10.1+dfsg-5+deb7u2	amd64 MIT Kerberos runtime libraries
libkrb5support0:amd64	1.10.1+dfsg-5+deb7u2	amd64 MIT Kerberos runtime libraries - Support library
libldap-2.4-2:amd64	2.4.31-1+nmu2	amd64 OpenLDAP libraries
liblocale-gettext-perl	1.05-7+b1	amd64 module using libc functions for internationalization in Perl
liblockfile-bin	1.09-5	amd64 support binaries for and cli utilities based on liblockfile
liblockfile1:amd64	1.09-5	amd64 NFS-safe locking library
liblzma5:amd64	5.1.1alpha+20120614-2	amd64 XZ-format compression library
liblzo2-2:amd64	2.06-1+deb7u1	amd64 data compression library
libmagic1:amd64	5.11-2+deb7u3	amd64 File type determination library using "magic" numbers
libmount1	2.20.1-5.3	amd64 block device id library
libmpc2:amd64	0.9-4	amd64 multiple precision complex floating-point library
libmpfr4:amd64	3.1.0-5	amd64 multiple precision floating-point computation
libncurses5:amd64	5.9-10	amd64 shared libraries for terminal handling
libncurses5-dev	5.9-10	amd64 developer's libraries for ncurses
libncursesw5:amd64	5.9-10	amd64 shared libraries for terminal handling (wide character support)
libnet-telnet-perl	3.03-3	All script telnetable connections
libnewt0.52	0.52.14-11.1	amd64 Not Erik's Windowing Toolkit - text mode windowing with slang
libnfnetlink0	1.0.0-1.1	amd64 Netfilter netlink library

Package Name	Version	Description
libonig2	5.9.1-1	amd64 Oniguruma regular expressions library
libopencryptoki0	2.3.1+dfsg-3	amd64 PKCS#11 implementation (library)
libp11-kit0:amd64	0.12-3	amd64 Library for loading and coordinating access to PKCS#11 modules - runtime
libpam-modules:amd64	1.1.3-7.1	amd64 Pluggable Authentication Modules for PAM
libpam-modules-bin	1.1.3-7.1	amd64 Pluggable Authentication Modules for PAM - helper binaries
libpam-runtime	1.1.3-7.1	All runtime support for the PAM library
libpam0g:amd64	1.1.3-7.1	amd64 Pluggable Authentication Modules library
libpcap0.8:amd64	1.3.0-1	amd64 system interface for user-level packet capture
libpci3:amd64	1:3.1.9-6	amd64 Linux PCI Utilities (shared library)
libpcre3:amd64	1:8.30-5	amd64 Perl 5 Compatible Regular Expression Library - runtime files
libperl-dev	5.14.2-21+deb7u1	amd64 Perl library: development files
libperl5.14	5.14.2-21+deb7u1	amd64 shared Perl library
libpipeline1:amd64	1.2.1-1	amd64 pipeline manipulation library
libpkcs11-helper1:amd64	1.09-1	amd64 library that simplifies the interaction with PKCS#11
libpopt0:amd64	1.16-7	amd64 lib for parsing cmdline parameters
libprocps0:amd64	1:3.3.3-3	amd64 library for accessing process information from /proc
libqdbm14	1.8.78-2	amd64 QDBM Database Libraries without GDBM wrapper[runtime]
libquadmath0:amd64	4.7.2-5	amd64 GCC Quad-Precision Math Library
libreadline6:amd64	6.2+dfsg-0.1	amd64 GNU readline and history libraries, run-time libraries
libsamplerate0:amd64	0.1.8-5	amd64 Audio sample rate conversion library
libsasl2-2:amd64	2.1.25.dfsg1-6+deb7u1	amd64 Cyrus SASL - authentication abstraction library
libsasl2-modules:amd64	2.1.25.dfsg1-6+deb7u1	amd64 Cyrus SASL - pluggable authentication modules
libselinux1:amd64	2.1.9-5	amd64 SELinux runtime shared libraries
libsemanage-common	2.1.6-6	All common files for SELinux policy management libraries
libsemanage1:amd64	2.1.6-6	amd64 SELinux policy management library
libsensors4:amd64	1:3.3.2-2+deb7u1	amd64 library to read temperature/voltage/fan sensors
libsepol1:amd64	2.1.4-3	amd64 SELinux library for manipulating binary security policies
libsigc++-2.0-0c2a:amd64	2.2.10-0.2	amd64 type-safe Signal Framework for C++ - runtime
libslang2:amd64	2.2.4-15	amd64 S-Lang programming library - runtime version
libsqlite3-0:amd64	3.7.13-1+deb7u1	amd64 SQLite 3 shared library
libsqlite3-dev	3.7.13-1+deb7u1	amd64 SQLite 3 development files
libss2:amd64	1.42.5-1.1	amd64 command-line interface parsing library
libssl1.0.0:amd64	1.0.1e-2+deb7u12	amd64 SSL shared libraries

Package Name	Version	Description
libstdc++6:amd64	4.7.2-5	amd64 GNU Standard C++ Library v3
libstdc++6-4.7-dev	4.7.2-5	amd64 GNU Standard C++ Library v3 (development files)
libswitch-perl	2.16-2	All switch statements for Perl
libtasn1-3:amd64	2.13-2	amd64 Manage ASN.1 structures (runtime)
libtext-charwidth-perl	0.04-7+b1	amd64 get display widths of characters on the terminal
libtext-iconv-perl	1.7-5	amd64 converts between character sets in Perl
libtext-wrapi18n-perl	0.06-7	All internationalized substitute of Text::Wrap
libtimedate-perl	1.2000-1	All collections of modules to manipulate date/time information
libtinfo-dev:amd64	5.9-10	amd64 developer's library for the low-level terminfo library
libtinfo5:amd64	5.9-10	amd64 shared low-level terminfo library for terminal handling
libtpm-unseal1	1.3.7-1	amd64 Management tools for the TPM hardware (library)
libtspi1	0.3.9-3+wheezy1	amd64 open-source TCG Software Stack (library)
libudev0:amd64	175-7.2	amd64 libudev shared library
libusb-0.1-4:amd64	2:0.1.12-20+nmu1	amd64 userspace USB programming library
libusb-1.0-0:amd64	2:1.0.11-1	amd64 userspace USB programming library
libustr-1.0-1:amd64	1.0.4-3	amd64 Micro string library: shared library
libuuid-perl	0.02-5	amd64 Perl extension for using UUID interfaces as defined in e2fsprogs
libuuid1:amd64	2.20.1-5.3	amd64 Universally Unique ID library
libwrap0:amd64	7.6.q-24	amd64 Wietse Venema's TCP wrappers library
libx11-6:amd64	2:1.5.0-1+deb7u1	amd64 X11 client-side library
libx11-data	2:1.5.0-1+deb7u1	all X11 client-side library
libx86-1:amd64	1.1+ds1-10	amd64 x86 real-mode library
libxapian22	1.2.12-2	amd64 Search engine library
libxau6:amd64	1:1.0.7-1	amd64 X11 authorisation library
libxcb1:amd64	1.8.1-2+deb7u1	amd64 X C Binding
libxdmcp6:amd64	1:1.1.1-1	amd64 X11 Display Manager Control Protocol library
libxext6:amd64	2:1.3.1-2+deb7u1	amd64 X11 miscellaneous extension library
libxml2:amd64	2.8.0+dfsg1-7+wheezy1	amd64 GNOME XML library
libxmuu1:amd64	2:1.1.1-1	amd64 X11 miscellaneous micro-utility library
linux-base	3.5	all Linux image base package
linux-compiler-gcc-4.6-x86	3.14.15-2~bpo70+1	amd64 Compiler for Linux on x86 (meta-package)
linux-headers-3.14-0.bpo.2-all-amd64	3.14.15-2~bpo70+1	amd64 All header files for Linux 3.14 (meta-package)
linux-headers-3.14-0.bpo.2-amd64	3.14.15-2~bpo70+1	amd64 Header files for Linux 3.14-0.bpo.2-amd64
linux-headers-3.14-0.bpo.2-common	3.14.15-2~bpo70+1	amd64 Common header files for Linux 3.14-0.bpo.2
linux-headers-3.14-0.bpo.2-common-rt	3.14.15-2~bpo70+1	amd64 Common header files for Linux 3.14-0.bpo.2-rt

Package Name	Version	Description
linux-headers-3.14-0.bpo.2-rt-amd64	3.14.15-2~bpo70+1	amd64 Header files for Linux 3.14-0.bpo.2-rt-amd64
linux-image-3.14-0.bpo.2-amd64	3.14.15-2~bpo70+1	amd64 Linux 3.14 for 64-bit PCs
linux-kbuild-3.14	3.14-1~bpo70+1	amd64 Kbuild infrastructure for Linux 3.14
linux-libc-dev:amd64	3.2.60-1+deb7u3	amd64 Linux support headers for userspace development
locales	2.13-38+deb7u4	all Embedded GNU C Library: National Language (locale) data [support]
lockfile-progs	0.1.17	amd64 Programs for locking and unlocking files and mailboxes
login	1:4.1.5.1-1	amd64 system login tools
logrotate	3.8.1-4	amd64 Log rotation utility
lsb-base	4.1+Debian8+deb7u1	All Linux Standard Base 4.1 init script functionality
make	3.81-8.2	amd64 An utility for Directing compilation.
man-db	2.6.2-1	amd64 on-line manual pager
manpages	3.44-1	All Manual pages about using a GNU/Linux system
manpages-dev	3.44-1	All Manual pages about using GNU/Linux for development
mawk	1.3.3-17	amd64 a pattern scanning and text processing language
mdadm	3.2.5-5	amd64 tool to administer Linux MD arrays (software RAID)
mime-support	3.52-1	All MIME files 'mime.types' & 'mailcap', and support programs
module-init-tools	9-3	All transitional dummy packages (module-init-tools to kmod)
mount	2.20.1-5.3	amd64 Tools for mounting and manipulating filesystems
multiarch-support	2.13-38+deb7u4	amd64 Transitional package to ensure multiarch compatibility
ncurses-base	5.9-10	All basic terminal type definitions
ncurses-bin	5.9-10	amd64 terminal-related programs and man pages
ncurses-term	5.9-10	All additional terminal type definitions
net-tools	1.60-24.2	amd64 The NET-3 networking toolkit
netbase	5.0	All basic TCP/IP networking systems
netcat-traditional	1.10-40	amd64 TCP/IP swiss army knife
ntpdate	1:4.2.6.p5+dfsg-2	amd64 client for setting system time from NTP servers
openbsd-inetd	0.20091229-2	amd64 OpenBSD Internet Superserver
opencryptoki	2.3.1+dfsg-3	amd64 PKCS#11 implementation (daemon)
openssh-blacklist	0.4.1+nmu1	All lists of default blacklisted OpenSSH RSA and DSA keys
openssh-blacklist-extra	0.4.1+nmu1	All lists of non-default blacklisted OpenSSH RSA and DSA keys
openssh-client	1:6.0p1-4+deb7u2	amd64 secure shell (SSH) client, for secure access to remote machines
openssh-server	1:6.0p1-4+deb7u2	amd64 secure shell (SSH) server, for secure access from remote machines

Package Name	Version	Description
openssl	1.0.1e-2+deb7u12	amd64 Secure Socket Layer (SSL) binary and related cryptographic tools
openvpn	2.2.1-8+deb7u2	amd64 virtual private network daemon
os-prober	1.58	amd64 utility to detect other OSes on a set of drives
passwd	1:4.1.5.1-1	amd64 change and administer password and group data
patch	2.6.1-3	amd64 Apply a diff file to an original
pciutils	1:3.1.9-6	amd64 Linux PCI Utilities
perl	5.14.2-21+deb7u1	amd64 Larry Wall's Practical Extraction and Report Language
perl-base	5.14.2-21+deb7u1	amd64 minimal Perl system
perl-modules	5.14.2-21+deb7u1	All core Perl modules
php5	5.4.4-14+deb7u12	All server-side, HTML-embedded scripting language (metapackage)
php5-cli	5.4.4-14+deb7u12	amd64 command-line interpreter for the php5 scripting language
php5-common	5.4.4-14+deb7u12	amd64 Common files for packages built from the php5 source
pm-utils	1.4.1-9	All utilities and scripts for power management
pmount	0.9.23-2	amd64 mount removable devices as normal user
powermgmt-base	1.31	amd64 Common utils and configs for power management
ppp	2.4.5-5.1+b1	amd64 Point-to-Point Protocol (PPP) - daemon
pppconfig	2.3.18+nmu4	All A text menu based utility for configuring ppp
pppoe	3.8-3	amd64 PPP over Ethernet driver
pppoeconf	1.20	All PPPoE/ADSL connections
procps	1:3.3.3-3	amd64 /proc file system utilities
proftpd-basic	1.3.4a-5+deb7u1	amd64 Versatile, virtual-hosting FTP daemon - binaries
proftpd-mod-vroot	0.9.2-2+b2	amd64 ProFTPD module mod_vroot
psmisc	22.19-1+deb7u1	amd64 utilities that use the proc file system
python	2.7.3-4+deb7u1	All interactive high-level object-oriented language (default version)
python-apt	0.8.8.2	amd64 Python interface to libapt-pkg
python-apt-common	0.8.8.2	All Python interfaces to libapt-pkg (locales)
python-chardet	2.0.1-2	All universal character encoding detectors
python-debian	0.1.21	All Python modules to work with Debian-related data formats
python-debianbts	1.11	All Python interfaces to Debian's Bug Tracking System
python-fpconst	0.7.2-5	All utilities for handling IEEE 754 floating point special values
python-minimal	2.7.3-4+deb7u1	All minimal subsets of the Python language (default version)
python-reportbug	6.4.4+deb7u1	All Python modules for interacting with bug tracking systems
python-soappy	0.12.0-4	All SOAP support for Python

Package Name	Version	Description
python-support	1.0.15	All automated rebuilding support for Python modules
python2.7	2.7.3-6+deb7u2	amd64 Interactive high-level object-oriented language (version 2.7)
python2.7-minimal	2.7.3-6+deb7u2	amd64 Minimal subset of the Python language (version 2.7)
readline-common	6.2+dfsg-0.1	All GNU readline and history libraries, common files
rsyslog	5.8.11-3	amd64 reliable system and kernel logging daemon
sed	4.2.1-10	amd64 The GNU sed stream editor
sensible-utils	0.0.7	All utilities for sensible alternative selection
sgml-base	1.26+nmu4	All SGML infrastructure and SGML catalog file support
sqlite3	3.7.13-1+deb7u1	amd64 Command line interface for SQLite 3
ssh	1:6.0p1-4+deb7u2	All secure shell client and server (metapackage)
ssl-cert	1.0.32	All simple debconf wrapper for OpenSSL
sudo	1.8.5p2-1+nmu1	amd64 Provide limited super user privileges to specific users
sysstat	10.0.5-1	amd64 system performance tools for Linux
sysv-rc	2.88dsf-41+deb7u1	All System-V-like runlevel change mechanism
sysvinit	2.88dsf-41+deb7u1	amd64 System-V-like init utilities
sysvinit-utils	2.88dsf-41+deb7u1	amd64 System-V-like utilities
tar	1.26+dfsg-0.1	amd64 GNU version of the tar archiving utility
tasksel	3.14.1	All Tool for selecting tasks for installation on Debian systems
tasksel-data	3.14.1	Al Official tasks used for installation of Debian systems
tcpd	7.6.q-24	amd64 Wietse Venema's TCP wrapper utilities
telnet	0.17-36	amd64 The telnet client
telnetd	0.17-36	amd64 The telnet server
tftpd	0.17-18	amd64 Trivial file transfer protocol server
tpm-tools	1.3.7-1	amd64 Management tools for the TPM hardware (tools)
traceroute	1:2.0.18-3	amd64 Traces the route taken by packets over an IPv4/IPv6 network
trousers	0.3.9-3+wheezy1	amd64 open-source TCG Software Stack (daemon)
tzdata	2014e-0wheezy1	All time zone and daylight-saving time data
ucf	3.0025+nmu3	All Update Configuration File: preserve user changes to config files.
udev	175-7.2	amd64 /dev/ and hotplug management daemon
update-inetd	4.43	All inetd configuration file updater
usbmount	0.0.22	All automatically mount and unmount USB mass storage devices
usbutils	1:005-3	amd64 Linux USB utilities
util-linux	2.20.1-5.3	amd64 Miscellaneous system utilities
vbetool	1.1-2	amd64 run real-mode video BIOS code to alter hardware state

Package Name	Version	Description
vim	2:7.3.547-7	amd64 Vi IMproved - enhanced vi editor
vim-common	2:7.3.547-7	amd64 Vi IMproved - Common files
vim-runtime	2:7.3.547-7	all Vi IMproved - Runtime files
vim-tiny	2:7.3.547-7	amd64 Vi IMproved - enhanced vi editor - compact version
watchdog	5.12-1	amd64 system health checker and software/hardware watchdog handler
wget	1.13.4-3+deb7u1	amd64 retrieves files from the web
whiptail	0.52.14-11.1	amd64 Displays user-friendly dialog boxes from shell scripts
xauth	1:1.0.7-1	amd64 X authentication utility
xkb-data	2.5.1-3	All X Keyboard Extension (XKB) configuration data
xml-core	0.13+nmu2	All XML infrastructure and XML catalog file support
xz-utils	5.1.1alpha+20120614-2	amd64 XZ-format compression utilities
zlib1g:amd64	1:1.2.7.dfsg-13	amd64 compression library - runtime