Application note

Commands and Scripts
OWL Family

Hirschmann Automation and Control GmbH
Stuttgarter Str. 45-51
72654 Neckartenzlingen
Germany

Contents

1 Commands and Scripts

arp

The arp program displays and modifies the Internet-to-Ethernet address translation tables used by the address resolution protocol.

Synopsis:

```
arp [-a <hostname>] [-s <hostname> <hw_addr>] [-d <hostname>] [-v] [-n] [-
i <if>] [-D <hostname>] [-A ] [-f <filename>]
```

Options:

Option	Description
-a	The entries will be displayed in alternate (BSD) style.
-s	Manually create an ARP address mapping entry for hostname with hardware address set to hw_addr.
-d	Remove any entry for the specified host.
-v	Tell the user what is going on by being verbose.
-n	Shows numerical addresses instead of trying to determine symbolic host, port or user names.
-i	Select an interface.
-D	Use the interface if as hardware address.
-f	Similar to the -s option, only with this option the address info is taken from file filename set up. The name of the data file is very often /etc/ ethers, but this is not official. If no filename is specified, /etc/ethers is used as default.The format of the file is simple; it only contains ASCII text lines with a hardware address and a hostname separated by whitespace. Additionally the pub, temp and netmask flags can be used

Table 1: arp options

With no flags, the program displays the current ARP entry for hostname. The host may be specified by name or by number, using Internet dot notation. For detail description of this command, visit Linux manual pages.

Examples:

View arp table without translating IP addresses to domain names

```
arp -n
```

awk

Awk scans each input file for lines that match any of a set of patterns specified literally in program-text or in 1 or more files specified as -f progfile.

Synopsis:

```
awk [-v] [-F] [-f] …[<program-text>] [<file> …]
```

Options:

Option	Description
-v	Assign the value `val` to the variable `var`, before execution of the program begins. Such variable values are available to the BEGIN block of an AWK program.
-F	Use for the input field separator (the value of the FS predefined variable).
-f	Read the AWK program source from the file program-file, instead of from the first command line argument. Multiple -f (or –file) options may be used.

Table 2: awk options

Examples:

Show IP address of Gateway

```
route -n | awk '/^0 .0 .0 .0/ { print $2 }'
```

brctl

The brctl command is used to set up, maintain, and inspect the Ethernet bridge configuration in the Linux kernel.

An Ethernet bridge is a device commonly used to connect different networks of Ethernets together, so that these Ethernets will appear as 1 Ethernet to the participants.

Each of the Ethernets being connected corresponds to 1 physical interface in the bridge. These individual Ethernets are bundled into 1 bigger (’logical’) Ethernet, this bigger Ethernet corresponds to the bridge network interface.

Synopsis:

```
brctl [<commands>]
```

Options:

Option	Parameters	Description
addbr	<bridge>	Add bridge
delbr	<bridge>	Delete bridge
addif	<bridge> <device>	Add interface to bridge
delif	<bridge> <device>	Delete interface from bridge
setageing	<bridge> <time>	Set aging time
setbridgepri	<bridge> <prio>	Set bridge priority
setfd	<bridge> <time>	Set bridge forward delay
sethello	<bridge> <time>	Set hello time
setmaxage	<bridge> <time>	Set max message age
setpathcost	<bridge> <port> <cost>	Set path cost
setportrpio	<bridge> <port> <prio>	Set port prioriy
show		Show list of bridges
showmacs	<bridge>	Show list of mac address
showstp	<bridge>	Show bridge stp info
stp	<bridge> {on | off}	Turn stp on/off

Table 3: brctl commands

Examples:

Create bridge between eth0 and eth1.

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
```

■ cat

This command concatenates files and print on the standard output.

Synopsis:

```
cat [-u] [<file>] …
```

Options:

Option	Description
-u	Ignored since unbuffered I/O is always used.

Table 4: cat options

Examples:

View the contents of file /proc/tty/driver/spear_serial (info about serial ports of v2 routers).

```
cat /proc/tty/driver/spear_serial
```

Copy the contents of the router configuration files in /tmp/my.cfg.

```
cat /etc/settings.* > /tmp/my.cfg
```

■ cd

This command is used to change the current working directory.

Synopsis:

```
cd [-P] [-L] [<directory>]
```

Options:

Option	Description
-P	Do not follow symbolic links
-L	Follow symbolic links (default)

Table 5: cd options

Examples:

Move to home directory (/root).

```
cd
```

Move to directory /mnt.

```
cd /mmt
```

cdmaat

The program used for sending AT command to CDMA module if available (equivalent of the gsmat command, See “gsmat” on page 19.)

Synopsis:

```
cdmaat <AT command>
```

cdmapwr

The program used to control the supply of CDMA module if available (equivalent of the gsmpwr command, See “gsmpwr” on page 21.)

Synopsis:

```
cdmapwr [on | off]
```

chmod

This command is used to change file mode bits.

Synopsis:

```
chmod [-R] <mode> <filename>
```

Options:

Option	Description
-R	Change files and directories recursively

Table 6: chmod options

Examples:

Settings rights (permit execution) of script /tmp/script.

```
chmod 755 /tmp/script
```

■ conntrack

This program is user interface to netfilter connection tracking system.

Synopsis:

```
conntrack [commands] [option]
```

Options:

Command	Description
-L [table] [option]	List conntrack or expectation table
-G [table]	Get conntrack or expectation
-D [table]	Delete conntrack or expectation
-I [table]	Create a conntrack or expectation
-U [table]	Update a conntrack
-E [table]	Show events
-F [table]	Flush table

Table 7: conntrack comands

Table	Description
conntrack	This is the default table. It contains a list of all currently tracked connections through the system.
expect	This is the table of expectations. Connection tracking expectations are the mechanism used to "expect" RELATED connections to existing ones.

Table 8: conntrack tables

Option	Description
-n <ip>	Source NAT ip
-g <ip>	Destination NAT ip
-m <mark>	Set mark
-e <eventmask>	Event mask, eg. NEW,DESTROY
-z	Zero counters while listing
-o <type[...]>	Output format, eg. xml

Table 9: conntrack options

Option	Description
--tuple-src <ip>	Source address in expect tuple
--tuple-dst <ip>	Destination address in expect tuple
--mask-src <ip>	Source mask address
--mask-dst <ip>	Destination mask address

Table 10: expectation options

Option	Description
-s <ip>	Source address from original direction
-d <ip>	Destination address from original direction
-r <ip>	Source addres from reply direction
-q <ip>	Destination address from reply direction
-p <proto>	Layer 4 Protocol, eg. ’tcp’
-f <proto>	Layer 3 Protocol, eg. ’ipv6’
-t <timeout>	Set timeout
-u <status>	Set status, eg. ASSURED

Table 11: conntrack and expectation options

Examples:

Display content of conntrack table.

```
conntrack -L
```

Delete content of contrack table.

```
conntrack -F
```

cp

This command is used to copy files and directories.

Synopsis:

```
cp [<option>] <source> <dest>
```

Options:

Option	Description
-a	Preserve the all attributes
-d, -P	Never follow symbolic links
-H, -L	Follow command-line symbolic links
-p	Preserve the mode, ownership, timestamps attributes
-f	If an existing destination file cannot be opened, remove it and try again
-i	Prompt before overwrite
-R, -r	Copy directories recursively

Table 12: cp options

Examples:

Copy the system log to directory /mnt.

```
cp /var/log/messages* /mnt
```

Copy configuration profile "Alternative 1" to profile "Standard".

```
cp -r /etc/alt1/* /etc
```

curl

Curl (transfer a URL) is a tool to transfer data from or to a server, using one of the supported protocols (DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, TELNET and TFTP). It is an alternative to wget .See “wget” on page 52.

Synopsis:

```
curl [options...] <url>
```

Options:
Type curl --help for options to show in the command line or visit online manual page at

```
http://curl.haxx.se/docs/manpage.html
```

date

This command is used to display the current time in the given FORMAT, or set the system date (and time).

Synopsis:

```
date [-R] [-d <string>] [-s] [-r <file>] [-u] [MMDDhhmm[[CC]YY][.ss]]
```

Options:

Option	Description
-R	Output date and time in RFC 2822 format
-d <string>	Display time described by STRING, not ’now’
-s	Set time described by STRING
-r <file>	Display the last modification time of FILE
-u	Print or set Coordinated Universal Time

Table 13: date options

Examples:

Display the current date and time.

```
date
```

Setting the date and time on December 24, 2011 20:00.

```
date 122420002011
```

defaults

The script is used to restore the default configuration.

Synopsis:

```
defaults
```

df

This command is used to view report file system disk space usage.

Synopsis:

```
df [-k] [<filesystem> …]
```

Options:

Option	Description
-k	Print sizes in kilobytes

Table 14: df options

dmesg

This command is used to print or control the kernel ring buffer.

Synopsis:

```
dmesg [-c] [-n <level>] [-s <size>]
```

Options:

Option	Description
-c	Clears the ring buffer’s contents after printing
-n <level>	Set the level at which logging of messages is done to the console
-s <size>	Use a buffer of size SIZE to query the kernel ring buffer. This is 16392 bydefault.

Table 15: dmesg options

Examples:

View the latest news and subsequent deletion of the kernel ring buffer.

```
dmesg -c
```

echo

This command prints the strings to standard output.

Synopsis:

```
echo [-n] [-e] [-E] [<string> ...]
```

Options:

Option	Description
-n	Do not output the trailing newline
-e <level>	Enable interpretation of backslash escapes
-E <size>	Disable interpretation of backslash escapes (default)

Table 16: echo options

Examples:

Switch profile to "Standard".

```
echo "PROFILE=" > /etc/settings
reboot
```

Switch profile to "Alternative 1".

```
echo "PROFILE=alt1" > /etc/settingsreboot
```

Send a sequence of bytes 0x41,0x54,0x0D,0x0A to serial line (write data in octal).

```
echo -n -e " 101 124 015 012" > /dev/ttyS0
```

email

The program used for sending email.

Synopsis:

```
email -t <to> [-s <subject>] [-m <message>] [-a <attachment>] [-r <retries>]
```

Options:

Option	Description
-t	Email of recipient
-s	Subject of email
-m	Message of email
-a	Attachment of email
-r	Number of retries

Table 17: email options

Examples:

Send system logs to the address john.doe@email.com.

```
email -t john.doe@email.com -s "System Log" -a /var/log/messages
```

ethtool

This command is used to display or change Ethernet card settings.

Synopsis:

```
ethtool [<option> …] <devname> [<commands>]
```

Options:

For detail description this command, visit Linux manual pages.

Examples:

View the status of the interface eth0.

```
ethtool eth0
```

Switch interface eth0 to mode 10 Mbit/s, half duplex.

```
ethtool -s eth0 speed 10 duplex half autoneg off
```

Turn on autonegacion on the interface eth0.

```
ethtool -s eth0 autoneg on
```

find

Command to search for files in a directory hierarchy.

Synopsis:
find [<path> …] [<expression>]

Options:
The default path is the current directory, default expression is ’-print’. Type find --help for help or look up online man page for more detailed description. Expression may consist of:

Option	Description
-follow	Dereference symbolic links
-name <pattern>	File name (leading directories removed) matches <pattern>
-print	Print (default and assumed)
-type X	Filetype matches X (where X is one of: f,d,l,b,c,…)
-perm <perms>	Permissions match any of (+NNN); all of (-NNN); or exactly (NNN)
-mtime <days>	Modified time is greater than (+N); less than (-N); or exactly (N) days
-mmin <mins>	Modified time is greater than (+N); less than (-N); or exactly (N) minutes
-exec <cmd>	Execute command with all instances of {} replaced by the files matching <expression>

Table 18: find expressions

Examples:

Search for files in your home directory which have been modified in the last 24 hours.

```
find $HOME -mtime 0
```

Search for files which have read and write permission for their owner, and group, but which other users can read but not write to. find

```
-perm 664
```

free

This command is used to display information about free and used memory.

Synopsis:

```
free
```

fwupdate

The program used for router’s firmware update.

Synopsis:
fwupdate [-i <filename> [-h] [-n]] [-f]

Options:

Option	Description
-i	File of the new firmware, filename has to be specified
-h	HTML output (used when called from web configuration)
-n	Do not reboot after firmware update
-f	finish update procedures, called by default

Table 19: fwupdate options

grep

Grep searches the named input FILEs (or standard input if no files are named, or the file name – is given) for lines containing a match to the given PATTERN. By default, grep prints the matching lines.

Synopsis:

```
grep [<options> …] <pattern> [<file> …]
```

Options:

Option	Description
-H	Print the filename for each match
-h	Suppress the prefixing of filenames on output when multiple files are searched
-i	Ignore case distinctions
-l	Suppress normal output; instead print the name of each input file from which output would normally have been printed
-L	Suppress normal output; instead print the name of each input file from which no output would normally have been printed
-n	Prefix each line of output with the line number within its input file
-q	Quiet; do not write anything to standard output. Exit immediately with zero status if any match is found, even if an error was detected. Also see the -s or --no-messages option.
-v	Invert the sense of matching, to select non-matching lines
-s	Suppress error messages about nonexistent or unreadable files
-c	Suppress normal output; instead print a count of matching lines for each input file
-f	Obtain patterns from FILE, one per line
-e	Use PATTERN as the pattern; useful to protect patterns beginning with –
-F	Interpret PATTERN as a list of fixed strings, separated by new lines, any of which is to be matched

Table 20: grep options

Examples:

See all lines of system log in which occurs the word "error".

```
grep error /var/log/messages
```

View all processes whose name the contents of the string "ppp".

```
ps | grep ppp
```

gsmat

The program used for sending AT command to GSM module.

Synopsis:

```
gsmat <AT command>
```

Examples:

Determine the type and firmware version of GSM module.

```
gsmat ATI
```

Determine the IMEI code of module.

```
gsmat "AT+GSN"
```

gsmat2

The program used for sending AT command to second GSM module if available.

Synopsis:

```
gsmat2 <AT command>
```

gsminfo

The program used to display information about the signal quality.

Synopsis:

Synopsis:
gsminfo

Options:

Option	Description
PLMN	Code of operator
Cell	The cell to which the router is connected
Channel	The channel on which the router communicates
Level	The signal quality of the selected cell
Neighbours	Signal quality of neighboring hearing cells
Uptime	Time to establish PPP connection

Table 21: Description of GSM information

gsmpwr

The program used to control the supply of GSM module.

Synopsis:

```
gsmpwr [on | off]
```

Examples:

Power of GSM module is turning on.

```
gsmpwr on
```

Power of GSM module is turning off.

```
gsmpwr off
```

gsmpwr2

The program used to control the supply of second GSM module if available.

Synopsis:

```
gsmpwr2 [on | off]
```

gsmsms

The program used to send SMS message.

Synopsis:

```
gsmsms <phone number> <text>
```

Examples:

Send SMS "Hello word" on telephone number +420123456789.

```
gsmsms +420123456789 "Hello word"
```

gunzip

This program is used to decompress FILE (or standard input if filename is '–').

Synopsis:

```
gunzip [-c] [-f] [-t] <filename>
```

Options:

Option	Description
-c	Write output on standard output
-f	Force decompression even if the file has multiple links or the corresp. filealready exists, or if the compressed data is read from or written to a terminal.
-t	Test. Check the compressed file integrity.

Table 22: gunzip options

Examples:

Decompression of file test.tar.gz (creates file test.tar).

```
gunzip test.tar.gz
```

gzip

This program is used to compress FILE with maximum compression.

Synopsis:

```
gzip [-c] [-d] [-f] <filename>
```

Options:

Option	Description
-c	Write output on standard output
-d	Decompress
-f	Force compression even if the file has multiple links or the corresponding file already exists, or if the compressed data is read from or written to a terminal

Table 23: gzip options

Examples:

Compression of file test.tar (creates file test.tar.gz).

```
gzip test.tar
```

hwclock

This program is used to query and set the hardware clock (RTC).

Synopsis:

```
hwclock [-r] [-s] [-w] [-u] [-l]
```

Options:

Option	Description
-r	Read hardware clock a print result
-s	Set the System Time from the Hardware Clock
-w	Set the Hardware Clock to the current System Time
-u	The hardware clock is kept in coordinated universal time
-l	The hardware clock is kept in local time

Table 24: hwclock options

Examples:

Set the hardware clock to the current system time.

```
hwclock -w -u
```

ifconfig

This command is used to configure a network interface.

Synopsis:

```
ifconfig [-a] <interface> [<option> …]
```

Options:

Option	Description
broadcast <addr.>	If the address argument is given, set the protocol broadcast address for this interface.
pointtopoint <ad.>	This keyword enables the point-to-point mode of an interface, meaning that it is a direct link between 2 machines with nobody else listening on it.
netmask <address>	Set the IP network mask for this interface.
dstaddr <address>	Set the remote IP address for a point-to-point link (such as PPP).
metric <NN>	This parameter sets the interface metric.
mtu <NN>	This parameter sets the Maximum Transfer Unit of an interface.
trailers	This flag used to cause a non-standard encapsulation of inet packets on certain link levels.
arp	Enable or disable the use of the ARP protocol on this interface.
allmulti	Enable or disable all-multicast mode. If selected, all multicast packets on the network will be received by the interface.
multicast	Set the multicast flag on the interface. This should not normally be needed as the drivers set the flag correctly them-selves.
promisc	Enable or disable the promiscuous mode of the interface. If selected, all packets on the network will be received by the interface.
txqueuelen <NN>	Set the length of the transmit queue of the device.
up | down	This flag causes the interface to be activated. | This flag causes the driver for this interface to be shut down.

Table 25: ifconfig options

Examples:

View the status of all interfaces.

```
ifconfig
```

Activation of loopback with IP address 127.0.0.1/8.

```
ifconfig lo up
```

Activation of virtual interface eth0:0 with IP address

```
192.168.2.1/24.ifconfig eth0:0 192.168.2.1 netmask 255.255.255.0 up
```

io

The program is used to control outputs and read inputs. Supports reading state of binary outputs and setting state of counters.

Synopsis:

```
io [get <pin>] | [set <pin> <value>]
```

Options:

Option	Description
get	Set output
set	Determine state of input

Table 26: io options

Examples:

Set the state of binary output OUT0 to 1.

```
io set out0 1
```

Determine the state of digital input BIN0.

```
io get bin0
```

Note: The `io get bin0` command returns a logical `0` if the corresponding digital input is set to a logical `1`.

Determine the state of analog input AN1 on expansion port XC-CNT.

```
io get an1
```

Determine the state of counter input CNT1 on expansion port XC-CNT.

```
io get cnt1
```

ip

This command is used to configure a network interface or show the current configuration. Type ip --help for help in the terminal.

The OWL routers support more ip options and commands (options: -d[etails] , -t[imestamp , -b[atch] <filename> , -rc[vbuf] ; objects: addrlabel , ntable , tuntap , mrule , netns , l2tp , tcp_metrics , token). For information how to use, type ip <object> help , for detailed description of all options, visit Linux manual pages or look up them online.

Synopsis:

```
ip [ <options> ] <object> { <command> | help }
```

Options:

Option	Description
-V[ersion]	Print the version of the ip utility and exit
-s[tatistics]	Output more information. If the option appears twice or more, the amount of information increases.
-r[esolve]	use the system's name resolver to print DNS names instead of host addresses
-f[amily] <family>	Specifies the protocol family to use. The protocol family identifier can be one of inet, inet6, bridge, ipx, dnet or link.
-o[neline]	output each record on a single line, replacing line feeds with the '\' character

Table 27: ip options

Object	Description
link	network device
addr	protocol (IP or IPv6) address on a device
route	routing table entry
rule	rule in routing policy database
neigh	manage ARP or NDISC cache entries
tunnel	tunnel over IP
maddr	multicast address
mroute	multicast routing cache entry
monitor	watch for netlink messages
xfrm	manage IPSec policies

Table 28: ip objects

Examples:

View the status of all interfaces.

```
ip link show
```

View the route table.

```
ip route list
```

Add routing networks 192.168.3.0/24 through interface eth0.

```
ip route add 192.168.3.0/24 dev eth0
```

Add routing IP address 192.168.3.1 trough gateway 192.168.1.2.

```
ip route add 192.168.3.1 via 192.168.1.2
```

Add default gateway 192.168.1.2.

```
ip route add default via 192.168.1.2
```

iptables

This command is used to administration tool for IP packet filtering and NAT.

Synopsis:

```
iptables [<options>]
```

Options:
For detail description of this command visit Linux manual pages.

Examples:

Redirect incoming TCP connections to port 8080 on IP address 192.168.1.2 and port 80.

```
iptables -t nat -A napt -p tcp --dport 8080 -j DNAT --to-destination
192.168.1.2:80
```

kill

This command is used to terminate process.

Synopsis:

```
kill [ -<signal> ] <process-id> [ <process-id> …]
kill -l
```

Options:

Option	Description
-l	Print a list of signal names. These are found in /usr/include/linux/signal.h
-q	Do not complain if no processes were killed

Table 29: kill options

Examples:

End the process with PID 1234 by sending signal SIGTERM.

```
kill 1234
```

End the process with PID 1234 by sending signal SIGKILL.

```
kill -9 1234
```

killall

This command is used to kill all process with process name.

Synopsis:

```
killall [ -q] [ -<signal> ] <process-name> [<process-name> …]
```

Options:

Option	Description
-l	Print a list of signal names. These are found in /usr/include/linux/signal.h
-q	Do not complain if no processes were killed

Table 30: killall options

Examples:

End the all processes with name pppd by sending signal SIGTERM.

```
killall pppd
```

End the all processes with name pppd by sending signal SIGKILL.

```
killall -9 pppd
```

led

The program used to control the USR LED on the front panel of the router.

Synopsis:

```
led [on | off]
```

Options:

Option	Description
on	User LED is on
off	User LED is off

Table 31: led options

Examples:

Turn on USR LED.

```
led on
```

Turn off USR LED.

```
led off
```

ln

The program used to make links between files.

Synopsis:

```
ln [ option ] < target > …< link_name > | < directory >
```

Options:

Option	Description
-s	Make symbolic links instead of hard links
-f	Remove existing destination files
-n	No dereference symlinks – treat like normal file
-b	Make a backup of the target (if exists) before link operation
-S	Use suffix instead of ~ when making backup files

Table 32: ln options

Examples:

Creating a symbolic link to file /var/log/messages called my.log.

```
ln -s /var/log/messages my.log
```

■ logger

The program makes entries in the system log. It provides a shell command interface to the system log module.

Synopsis:

```
logger [ option ] [ message …]
```

Options:

Option	Description
-i	Log the process id of the logger process with each line
-s	Log the message to standard error, as well as the system log
-f <file>	Log the specified file
-p <priority>	Enter the message with the specified priority. The priority may be specified numerically or as a facility.level pair.
-t <tag>	Mark every line in the log with the specified tag
-u <socket>	Write to socket as specified with socket instead of builtin syslog routines
-d	Use a datagram instead of a stream connection to this socket

Table 33: logger options

Examples:

Send the message System rebooted to the syslogd daemon.

```
logger System rebooted
```

Send the message System going down immediately!!! to the syslog daemon, at the emerg level and user facility.

```
logger -p user.emerg "System going down immediately!!!
```

lpm

Put the router into the low power mode and wake up on events specified by parameters (binary input or time interval). Router will wake up on the first event coming when more parameters specified.

This command works on OWL routers only due to hardware support.

Synopsis:

Synopsis:
lpm [-b] [-i <interval>]

Options:

Option	Description
-b	Wake up the router on binary input In1
-i	Wake up the router after time interval specified in seconds

Table 34: lpm options

ls

The program used to list directory contents.

Synopsis:

```
ls [ option ] < filename > …
```

Options:

Option	Description
-1	List files in a single column
-A	Do not list implied . and ..
-a	Do not hide entries starting with .
-C	List entries by columns
-c	With -l: show ctime
-d	List directory entries instead of contents
-e	List both full date and full time
-i	List the i-node for each file
-l	Use a long listing form
-n	List numeric UIDs and GIDs instead of names
-L	List entries pointed to by symbolic links
-r	Sort the listing in reverse order
-S	Sort the listing by file size
-s	List the size of each file, in blocks
-t	With -l: show modification time
-u	With -l: show access time
-v	Sort the listing by version
-x	List entries by lines instead of by columns
-X	Sort the listing by extension

Table 35: ls options

Examples:

View list contents of actually directory.

```
ls
```

mac

The program used to display the MAC address of eth0.

Synopsis:

```
mac [<separator>]
```

Examples:

Display the MAC address of eth0. Will be used as the separator character "-" instead of ":".

```
mac -
```

mkdir

This program used to make directories.

Synopsis:

Synopsis:
mkdir [<option>] directory …

Options:

Option	Description
-m	Set permission mode (as in chmod), not rwxrwxrwx – umask
-p	No error if existing, make parent directories as needed

Table 36: mkdir options

Examples:

```
mkdir -p /tmp/test/example
```

mount

This program used to mount a file system.

Synopsis:

```
mount [-a] [-o] [-r] [-t] [-w] <DEVICE> <NODE> [ -o <option>, …]
```

Options:

Flag	Description
-a	Mount all filesystems in fstab
-o	one of many filesystem options, listed below
-r	Mount the filesystem read-only
-t	Specify the filesystem type
-w	Mount for reading and writing (default)

Table 37: mount flags

Option	Description
async/sync	Writes are asynchronous/synchronous
atime/noatime	Enable/disable updates to inode access times
dev/nodev	Allow use of special device files/disallow them
exec/noexec	Allow use of executable files/disallow them
suid/nosuid	Allow set-user-id-root programs/disallow them
remount	Re-mount a mounted filesystem, changing its flags
ro/rw	Mount for read-only/read-write
bind	Bind a directory to an additional location
move	Relocate an existing mount point

Table 38: mount options

For detail description this command, visit Linux manual pages.

Examples:

Connect a contents of USB flash drive to the directory /mnt.

```
mount -t vfat /dev/sda1 /mnt
```

mv

This program is used to move or rename files.

Synopsis:

```
mv [-f] [-i] <source> …<dest>
```

Options:

Option	Description
-f	Don’t prompt before overwriting
-i	Interactive, prompt before overwrite

Table 39: mv options

Examples:

Rename file abc.txt na def.txt.

```
mv abc.txt def.txt
```

Move all files with the extension txt to the directory /mnt.

```
mv *.txt /mnt
```

nc

This program Netcat opens a pipe to IP:port.

Synopsis:

```
nc [<options>] [<ip>] [<port>]
```

Options:

Option	Description
-l	listen mode, for inbound connects
-p <port>	local port number
-i <secs>	delay interval for lines sent
-w <secs>	timeout for connects and final net reads

Table 40: nc options

Examples:

Open a TCP connection to port 42 of 192.168.3.1, using port 31337 as the source port, with a timeout of 5 seconds:

```
nc -p 31337 -w 5 192.168.3.1 42
```

■ netstat

The program Netstat displays the networking information.

Synopsis:

```
netstat [<options>]
```

Options:

Option	Description
-l	display listening server sockets
-a	display all sockets (default: connected)
-e	display other/more information
-n	don’t resolve names
-r	display routing table
-t	tcp sockets
-u	udp sockets
-w	raw sockets
-x	unix sockets

Table 41: netstat options

■ ntpdate

The program is used to set the system time from NTP server.

Synopsis:

```
ntpdate [-p <probes>] [-t <timeout>] <server>
```

Options:

Option	Description
-p	Specify the number of samples to be acquired from each server as the integer samples, with values from 1 to 8 inclusive.
-t	Specify the maximum time waiting for a server response as the value timeout, in seconds and fraction.

Table 42: ntpdate options

Examples:

Set the system time according to the NTP server time.windows.com.

```
ntpdate time.windows.com
```

openssl

The openssl program is a command line tool for using the various cryptography functions of OpenSSL's crypto library from the shell. It can be used for:

- Creation of RSA, DH and DSA key parameters
- Creation of X.509 certificates, CSRs and CRLs
- Calculation of Message Digests
- Encryption and Decryption with Ciphers
- SSL/TLS Client and Server Tests
- Handling of S/MIME signed or encrypted mail

Synopsis:

```
openssl [<option> …]
```

Options:

For detail description this command, visit Linux manual pages.

Examples:

Generate a new key for the SSH server.

```
openssl genrsa -out /etc/certs/ssh_rsa_key 512
```

Generate a new certificate for the HTTPS server.

```
openssl req -new -out /tmp/csr -newkey rsa:1024 -nodes -keyout /etc/certs/
https_key
openssl x509 -req -setstart 700101000000Z -setend 400101000000Z -in /tmp/
csr -signkey /etc/certs/https_key -out /etc/certs/https_cert
```

■ passwd

This program is used to change password for user root.

Synopsis:

```
passwd
```

■ pidof

This program lists the PIDs of all processes with names that match the names on the command line.

Synopsis:

```
pidof <process-name> [<option>] [<process-name> ...]
```

Options:

Option	Description
-s	display only a single PID

Table 43: pidof options

■ ping

This program is used to send ICMP echo request to network host.

Synopsis:

```
ping [-c <count>] [-s <size>] [-q] <hosts>
```

Options:

Option	Description
-c	Send only COUNT pings
-s	Send SIZE data bytes in packets (default = 56)
-q	Quiet mode, only displays output at start and when finished
-I	Selects outgoing interface

Table 44: ping options

Examples:

Send 1 ICMP packet Echo Request with size 500 B on IP address 10.0.0.1.

```
ping -c 1 -s 500 10.0.0.1
```

portd

The program is used for transparent transfer of data from the serial line by TCP or UDP.

Synopsis:

```
[-l <split timeout>] [-4] [-h <hostname>] [-o <proto>] -t <port> [-k
<keepalive time>] [-i <keepalive interval>] [-r <keepalive probes>] [-x] [-
z]
portd -c <device> [-b <baudrate>] [-d <databits>] [-p <parity>] [-s
<stopbits>]
```

Options:

Option	Description
-c	Serial line device
-b	Baudrate
-d	Number of data bits
-p	Parity – even, odd or none
-s	Number of stop bits
-l	Split timeout
-4	Forced detection Expansion port 485
-h	Hostname
	Protocol TCP or UDP
-t	TCP or UDP port
-k	Keepalive time
-i	Keepalive interval
-r	Keepalive probes
-x	Use signal CD as indicator of the TCP connection
-z	Use DTR as control TCP connection

Table 45: portd options

Examples:

Running a TCP server listening on port 1000th After a TCP connection, the program transparently transmit data from the serial port settings 115200 bit/s, 8N1.

```
portd -c /dev/ttyS0 -b 115200 -t 1000 &
```

ps

This program is used to view report process status.

Synopsis:

```
ps
```

pwd

This program used to view current directory.

Synopsis:

```
pwd
```

reboot

This program is used to reboot the router.

Synopsis:

```
reboot [-d <delay>] [-n <nosync>] [-f <force>]
```

Options:

Option	Description
-d	Delay interval for rebooting
-n	No call to sync()
-f	Force reboot, do not call shutdown

Table 46: reboot options

Examples:

Reboot router after 10 second.

```
reboot -d 10
```

restore

This program is used to restore configuration from file.

Synopsis:

```
restore <filename>
```

Examples:

Restore configuration from file /tmp/my.cfg.

```
restore /tmp/my.cfg
```

rm

This program is used to remove files or directories.

Synopsis:

```
rm [-i] [-f] [-r] <file> …
```

Options:

Option	Description
-i	Always prompt before removing each destination
-f	Remove existing destinations, never prompt
-r	Remove the contents of directories recursively

Table 47: rm options

Examples:

Remove all files with extension txt in the current directory.

```
rm *.txt
```

Remove directory /tmp/test and all subdirectories.

```
rm -rf /tmp/test
```

rmdir

This program is used to remove empty directories.

Synopsis:

```
rmdir <filename>
```

Examples:

Remove empty directory /tmp/test.

```
rmdir /tmp/test
```

route

This program is used to show and manipulate the IP routing table.

Synopsis:

```
route [ -n ] [ -e ] [ -A ] [ add | del | delete ]
```

Options:

Option	Description
-n	Don’t resolve names
-e	Display other/more information
-A	Select address family

Table 48: route options

For detail description this command, visit Linux manual pages.

Examples:

View the routing table without translating IP addresses to domain names.

```
route -n
```

Add routing networks 192.168.3.0/24 through eth0.

```
route add -net 192.168.3.0/24 dev eth0
```

Add routing IP addresses 192.168.3.1 through 192.168.1.2 gateway.

```
route add -host 192.168.3.1 gw 192.168.1.2
```

Add default gateway 192.168.1.2

```
route add default gw 192.168.1.2
```

sed

This program is used for filtering and transforming text.

Synopsis:

```
sed [ -e ] [ -f ] [ -i ] [ -n ] [ -r ] pattern [ -files ]
```

Options:

Option	Description
-e	Add the script to the commands to be executed
-f	Add script-file contents to the commands to be executed
-i	Edit files in place (makes backup if extension supplied)
-n	Suppress automatic printing of pattern space
-r	Use extended regular expression syntax

Table 49: sed options

If no -e or -f is given, the first non-option argument is taken as the sed script to interpret. All remaining arguments are names of input files; if no input files are specified, then the standard input is read. Source files will not be modified unless -i option is given.

Examples:

Change parameter PPP_APN in file /etc/settings.ppp to value "internet".

```
sed -e "s/ (PPP_APN= ).*/ 1internet/" -i /etc/settings.ppp
```

service

This program is used to start, stop or restart specified service.

Synopsis:

```
service < service name > <start | stop | restart>
```

Examples:

Start service cron.

```
service cron start
```

Restart service ppp.

```
service ppp restart
```

■ sleep

This program is used to delay for a specified amount of time.

Synopsis:

```
sleep <time>
```

Examples:

Sleep for 30 seconds.

```
sleep 30
```

■ slog

This script used to show system log (file /var/log/message).

Synopsis:

```
slog [-n <number>] [-f]
```

Options:

Option	Description
-n	Print last N lines instead of last 10
-f	Output data as the file grows

Table 50: slog options

Examples:

Continuous listing the system log. Listing stops when reaching the maximum number of lines of log.

```
slog -f
```

■ snmptrap

This program is used to sending SNMP trap.

Synopsis:

```
snmptrap [-c <community>] [-g <generic>] [-s <specific>] <hostname> [<oid>
<type> <value>]
```

Options:

Option	Description
-c	Community
-g	Specifies generic trap types: ▶ 0 – coldStart ▶ 1 – warmStart ▶ 2 – linkDown ▶ 3 – linkUp ▶ 4 – authenticationFailure ▶ 5 – egpNeighborLoss ▶ 6 – enterpriseSpecific
-r	Sends MAC address of eth0 interface
-s	Specifies user definition trap types in the enterpriseSpecific

Table 51: snmptrap options

Examples:

Send TRAP with info about the status of a digital input BIN0 to the IP address 192.168.1.2.

```
snmptrap 192.168.1.2 1.3.6.1.4.1.30140.2.3.1.0 u ’io get bin0’
```

Send TRAP "warm start" to the IP address 192.168.1.2

```
snmptrap -g 1 192.168.1.2
```

■ status

This program writes out the status of router’s interfaces or system. It is equivalent to General Status and Mobile WAN Status in router’s web administration.

Synopsis:

```
status [ -h ] [ -v ] [ lan | mobile | module | ports | ppp | sys | wifi ]
```

Options:

Option	Description
-h	Generates html output (used when called by web interface)
-v	Verbose – writes out more detailed informations
lan	Status of primary LAN. Can be lan 1, lan 2, etc. if available
mobile	Status of mobile WAN
module	Status of mobile module. Can be module 1, module 2, etc. if available
ports	Status of available peripheral ports
ppp	Status of mobile connection
sys	System information
wifi	Status of wlan interface

Table 52: status options

Examples:

Show verbosed status of mobile connection.

```
status -v mobile
```

tail

This program is used to output the last part of files.

Synopsis:

```
tail [ -n <number>] [ -f ]
```

Options:

Option	Description
-n	Print last N lines instead of last 10
-f	Output data as the file grows

Table 53: tail options

Examples:

Show last 30 lines of /var/log/messages.

```
tail -n 30 /var/log/messages
```

tar

This program is used to create, extract or list files from a tar file.

Synopsis:

```
tar -[czxtv0] [ -f tarfile ] [ -C dir ] [ file ] …
```

Options:

Option	Description
c	Create
x	Extract
t	List
z	Filter the archive trough gzip
-f	Name of TARFILE or "-" for stdin
0	Extract to stdout
-C	Change to directory DIR before operation
v	Verbosely list files processed

Table 54: tar options

Examples:

Creating log.tar archive that contains files from the directory /var/log.

```
tar -cf log.tar /var/log
```

Extract files from the archive log.tar.

```
tar -xf log.tar
```

tcpdump

This program is used to dump traffic on a network.

Synopsis:

```
tcpdump [-AdDeflLnNOpqRStuUvxX] [-c <count>] [-C <file size>]
[-E algo:secret][-F <file>] [-i <interface>] [-r <file>]
[-s <snaplen>] [-T type] [-w <file>][-y <datalinktype>] [expression]
```

Options:

For detail description this command, visit Linux manual pages.

Examples:

View traffic on interface usb0.

```
tcpdump -n -i usb0
```

View traffic on interface eth0 except protocol Telnet.

```
tcpdump -n not tcp port 23
```

View UDP traffic on interface eth0.

```
tcpdump -n udp
```

View HTTP traffic on interface eth0.

```
tcpdump -n tcp port 80
```

View all traffic from/to IP address 192.168.1.2.

```
tcpdump -n host 192.168.1.2
```

View traffic from/to IP address 192.168.1.2 except protocol Telnet.

```
tcpdump -n host 192.168.1.2 and not tcp port 23
```

telnet

This program is used to establish interactive communication with another computer over a network using the TELNET protocol.

Synopsis:

```
telnet <host> [<port>]
```

Examples:

Connect to 192.168.1.2 by protocol Telnet.

```
telnet 192.168.1.2
```

touch

This program used to update timestamp of file.

Synopsis:

```
touch [-c] <file> [<file> …]
```

Options:

Option	Description
-c	Do not create any files

Table 55: touch options

Examples:

Create a file, respectively update timestamp of file /tmp/test.

```
touch /tmp/test
```

traceroute

This program is printed the route packets trace to network host.

Synopsis:

```
traceroute [-FIldnrv] [-f <1st_ttl>] [-m <max_ttl>] [-p <port#>] [-q
<nqueries>] [-s <src_addr>] [-t <tos>] [-w <wait>] [-g <gateway>] [-i
<iface>] [-z <pausemsecs>] host [data size]
```

Options:

Option	Description
-F	Set the don’t fragment bit
-I	Use ICMP ECHO instead of UDP datagrams
-l	Display the ttl value of the returned packet
-d	Enable socket level debugging
-n	Print hop addresses numerically rather than symbolically
-r	Bypass the normal routing tables and send directly to a host
-v	Verbose output
-m	Set the max time-to-live (max number of hops)
-p	Set the base UDP port number used in probes (default is 33434)
-q	Set the number of probes per "ttl" to nqueries (default is 3)
-s	Use the following IP address as the source address
-t	Set the type-of-service in probe packets to the following value (default 0)
-w	Set the time (in seconds) to wait for a response to a probe (default 3 sec)
-g	Specify a loose source route gateway (8 maximum)

Table 56: traceroute options

Example:

Start traceroute on IP address 10.0.0.1 (without translation IP addresses to domain names).

```
traceroute -n 10.0.0.1
```

umount

This program is used to umount file systems.

Synopsis:

```
umount [-a] [-r] [-l] [-f] <file system> | <directory>
```

Options:

Option	Description
-a	Unmount all file systems
-r	Try to remount devices as read-only if mount is busy
-l	Lazy umount (detach filesystem)
-f	Force umount (i.e. unreachable NFS server)

Table 57: umount options

Examples:

Disconnecting the disc connected to the directory /mnt.

```
umount /mnt
```

vi

This program is used to edit and read text file.

Synopsis:

```
vi [-R] [<file> …]
```

Options:

Option	Description
-R	Read only, do not write to the file

Table 58: vi options

Examples:

Open file /etc/rc.local in the text editor vi.

```
vi /etc/rc.local
```

wget

This program is used to retrieve files via HTTP or FTP.

Synopsis:

```
wget [-c] [-q] [-O <document file>] [--header ’header: value’]
[-Y on/off] [-P <DIR>] <url>
```

Options:

Option	Description
-c	Continue retrieval of aborted transfers
-q	Quiet mode – do not print
-P	Set directory prefix to DIR
-O	Save to filename (’-’ for stdout)
-Y	Use proxy (’on’ or ’off’)

Table 59: wget options

Examples:

Download a file my.cfg from HTTP server with IP address 10.0.0.1.

```
wget http://10.0.0.1/my.cfg
```

xargs

This program executes the command on every item given by standard input.

Synopsis:

```
xargs [<commands>] [<options>] [<args> ...]
```

Options:

Option	Description
-r	Do not run command for empty readed lines
-t	Print the command line on stderr before executing it

Table 60: xargs options

Examples:

Find files named core in or below the directory /tmp and delete them. Note that this will work incorrectly if there are any filenames containing newlines or spaces.

```
find /tmp -name core -type f -print | xargs /bin/rm -f
```

1.1 Examples of scripts

1.1.1 Send SMS to email

Send incoming SMS to the email.

Startup Script:

```
EMAIL=john.doe@email.com
cat > /var/scripts/sms << EOF
#!/bin/sh
/usr/bin/email -t \$EMAIL -s "Received SMS from \$2" -m "Authorized:
    \$1, Text: \$3 \$4 \$5 \$6 \$7 \$8"
EOF
```

1.1.2 SMS command 1

Implementation of a new SMS command "IMPULSE", which activates binary output OUT0 for 5 seconds. SMS will be processed, if it comes from 1 of 3 numbers defined on the web interface or phone number +490123456789.

Startup Script:

```
PHONE=+490123456789
cat > /var/scripts/sms << EOF
#!/bin/sh
if [ "\$1" = "1" ] || [ "\$2" = "$PHONE" ]; then
 if [ "\$3" = "IMPULSE" ]; then
  /usr/bin/io set out0 1
  sleep 5
  /usr/bin/io set out0 0
 fi
fi
EOF
```

1.1.3 SMS command 2

This script implements a new SMS command "PPP", which sets item Network type , Default SIM card and Backup SIM card . PPP command has the following structure:

PPP <AUTO/GPRS/UMTS> <1/2>

The first parameter sets network type. If the second parameter equals 1, Default SIM card will be set to primary SIM card. If this parameter equals 2, Default SIM card will be set to secondary SIM card.

Startup Script:

```
cat > /var/scripts/sms << EOF
STARTUP=#!/bin/sh
if [ "\$1" = "1" ]; then
 if [ "\$3" = "PPP" ]; then
  if [ "\$4" = "AUTO" ]; then
   sed -e "s/\(PPP_NETTYPE=\).*/\10/" -e "s/   \(PPP_NETTYPE2=\).*/
   \10/" -i /etc/settings.ppp
  elif [ "\$4" = "GPRS" ]; then
   sed -e "s/\(PPP_NETTYPE=\).*/\11/" -e "s/   \(PPP_NETTYPE2=\).*/
   \11/" -i /etc/settings.ppp
  elif [ "\$4" = "UMTS" ]; then
   sed -e "s/\(PPP_NETTYPE=\).*/\12/" -e "s/   \(PPP_NETTYPE2=\).*/
   \12/" -i /etc/settings.ppp
  fi
  if [ "\$5" = "1" ]; then
   sed -e "s/\(PPP_DEFAULT_SIM=\).*/\11/" -e "s/
   \(PPP_BACKUP_SIM=\).*/\12/" -i /etc/settings.ppp
  elif [ "\$5" = "2" ]; then
   sed -e "s/\(PPP_DEFAULT_SIM=\).*/\12/" -e "s/
   \(PPP_BACKUP_SIM=\).*/\11/" -i /etc/settings.ppp
  fi
  reboot
 fi
fi
EOF
```

1.1.4 Send information email 1

Send information email about establishing of PPP connection.

Up Script:

```
EMAIL=john.doe@email.com
/usr/bin/email -t $EMAIL -s "Router has established PPP connection.
   IP address: $4"
```

1.1.5 Send information SNMP trap 1

Send information SNMP trap about establishing of PPP connection.

Up Script:

```
SNMP_MANAGER=192.168.1.2
/usr/bin/snmptrap -g 3 $SNMP_MANAGER
```

1.1.6 Send information email 2

Send information email about switch binary input BIN0.

Startup Script:

```
EMAIL=john.doe@email.com
MESSAGE="BIN0 is active"

while true
do
  /usr/bin/io get bin0
  VAL=$?
  if [ "$VAL" != "$OLD" ]; then
    [ "$VAL" = "0" ] && /usr/bin/email -t $EMAIL -s   "$MESSAGE"
    OLD=$VAL
  fi
  sleep 1
done
```

1.1.7 Send information SNMP trap 2

Send information SNMP trap about change state of binary input BIN0.

Startup Script:

```
SNMP_MANAGER=192.168.1.2

while true
do
  /usr/bin/io get bin0
  VAL=$?
  if [ "$VAL" != "$OLD" ]; then
     /usr/bin/snmptrap $SNMP_MANAGER 1.3.6.1.4.1.30140.2.3.1.0 u $VAL
     OLD=$VAL
  fi
  sleep 1
done
```

1.1.8 Automatic reboot

Automatic reboot at the definition time. (23:55)

Startup Script:

```
echo "55 23 * * * root /sbin/reboot" > /etc/crontab
service cron start
```

1.1.9 Switch between WAN and PPP

Switching between WAN and PPP. PPP connection is active, if PING on the defined IP address does not pass through.

Startup Script:

```
WAN_PING=192.168.2.1
WAN_GATEWAY=192.168.2.1
WAN_DNS=192.168.2.1

. /etc/settings.eth

/sbin/route add $WAN_PING gw $WAN_GATEWAY
/sbin/iptables -t nat -A PREROUTING -i eth1 -j napt
/sbin/iptables -t nat -A POSTROUTING -o eth1 -p ! esp -j MASQUERADE

LAST=1
while true
do
  ping -c 1 $WAN_PING
  PING=$?
  if [ $PING != $LAST ]; then
    LAST=$PING
    if [ $PING = 0 ]; then
      /etc/init.d/ppp stop
      sleep 3
     /sbin/route add default gw $WAN_GATEWAY
     echo "nameserver $WAN_DNS" > /etc/resolv.conf
     /usr/sbin/conntrack -F
     /etc/scripts/ip-up - - - $ETH2_IPADDR
    else
     /etc/scripts/ip-down - - - $ETH2_IPADDR
     /usr/sbin/conntrack -F
     /sbin/route del default gw $WAN_GATEWAY
     /etc/init.d/ppp start
    fi
  fi
  sleep 1
done
```

1.1.10 Add more MAC addresses reservation to DHCP server

At first, it is necessary to edit eth file (/etc/rc.d/init.d/eth) in a way that is illustrated below (marked lines).

```
#!/bin/sh
. /etc/settings
. /etc/$PROFILE/settings.eth
. /etc/$PROFILE/settings.ppp
. /root/DHCP_MAC
case "$1" in start|restart) echo -n "Setting up network: "
.
:
fi
if [ "$ETH_DHCP_STAT_ENABLED" = "1" ]; then [ -n "$ETH_DHCP_STAT_MAC1" ]
    && [ -n "$ETH_DHCP_STAT_IPADDR1" ] && HOST1="\\nhost 1
    { hardware ethernet $ETH_DHCP_STAT_MAC1; fixed-address
    $ETH_DHCP_STAT_IPADDR1; }"
    [ -n "$ETH_DHCP_STAT_MAC2" ] && [ -n "$ETH_DHCP_STAT_IPADDR2" ]
    && HOST2="\\nhost 2
    { hardware ethernet $ETH_DHCP_STAT_MAC2; fixed-address
    $ETH_DHCP_STAT_IPADDR2; }"
    [ -n "$ETH_DHCP_STAT_MAC3" ] && [ -n "$ETH_DHCP_STAT_IPADDR3" ]
    && HOST3="\\nhost 3
    { hardware ethernet $ETH_DHCP_STAT_MAC3; fixed-address
    $ETH_DHCP_STAT_IPADDR3; }"
    [ -n "$ETH_DHCP_STAT_MAC4" ] && [ -n "$ETH_DHCP_STAT_IPADDR4" ]
    && HOST4="\\nhost 4
    { hardware ethernet $ETH_DHCP_STAT_MAC4; fixed-address
    $ETH_DHCP_STAT_IPADDR4; }"
    [ -n "$ETH_DHCP_STAT_MAC5" ] && [ -n "$ETH_DHCP_STAT_IPADDR5" ]
    && HOST5="\\nhost 5 { hardware ethernet $ETH_DHCP_STAT_MAC5;
    fixed-address $ETH_DHCP_STAT_IPADDR5; }"
    [ -n "$ETH_DHCP_STAT_MAC6" ] && [ -n "$ETH_DHCP_STAT_IPADDR6" ]
    && HOST6="\\nhost 6
    { hardware ethernet $ETH_DHCP_STAT_MAC6; fixed-address
    $ETH_DHCP_STAT_IPADDR6; }"
    [ -n "$ETH_DHCP_STAT_MAC7" ] && [ -n "$ETH_DHCP_STAT_IPADDR7" ]
    && HOST7="\\nhost 7    { hardware ethernet $ETH_DHCP_STAT_MAC7; fixed-
address
    $ETH_DHCP_STAT_IPADDR7; }"    [ -n "$ETH_DHCP_STAT_MAC8" ] && [ -n
"$ETH_DHCP_STAT_IPADDR8" ]
    && HOST8="\\nhost 8    { hardware ethernet $ETH_DHCP_STAT_MAC8; fixed-
address
    $ETH_DHCP_STAT_IPADDR8; }"    [ -n "$ETH_DHCP_STAT_MAC9" ] && [ -n
"$ETH_DHCP_STAT_IPADDR9" ]
    && HOST9="\\nhost 9    { hardware ethernet $ETH_DHCP_STAT_MAC9; fixed-
address
    $ETH_DHCP_STAT_IPADDR9; }"
.
:
fi
```

```
echo -e "option routers $ETH_IPADDR;" \
 "\\noption domain-name-servers $ETH_IPADDR;" \
 "\\ndefault-lease-time $ETH_DHCP_LEASE_TIME;" \
 "\\nmax-lease-time 86400;" \
 "\\nsubnet $ETH_NETWORK netmask $ETH_NETMASK { $POOL }" \
 "$HOST1$HOST2$HOST3$HOST4$HOST5$HOST6$HOST7$HOST8$HOST9" >
 /var/dhcp/dhcpd.conf
touch /var/dhcp/dhcpd.leases
 /usr/sbin/dhcpd -q -cf /var/dhcp/dhcpd.conf -lf
 /var/dhcp/dhcpd.leases $ETH_IFNAME 2>
 /dev/null & if [ $? = 0 ]; then echo
 "done"; else echo "failed"; fi exit 0
```

Create a file named DHCP_MAC and copy it to folder /root/. It is possible to edit this file (/root/DHCP_MAC) as you need (MAC addresses and IP addresses). Finally, reboot router or press Apply button on LAN page in the web interface of your router.

Example of DHCP_MAC file:

```
ETH_DHCP_STAT_MAC7=00:0A:14:80:92:2F ETH_DHCP_STAT_IPADDR7=192.168.1.55

ETH_DHCP_STAT_MAC8=00:0A:14:12:34:56 ETH_DHCP_STAT_IPADDR8=192.168.1.11

ETH_DHCP_STAT_MAC9=00:0A:14:F0:92:6A ETH_DHCP_STAT_IPADDR9=192.168.1.71
```

A Further support

Technical questions

For technical questions, please contact any Hirschmann dealer in your area or Hirschmann directly.

You find the addresses of our partners on the Internet at http://www.hirschmann.com.

A list of local telephone numbers and email addresses for technical support directly from Hirschmann is available at https://hirschmann-support.belden.com.

This site also includes a free of charge knowledge base and a software download section.

Customer Innovation Center

The Customer Innovation Center is ahead of its competitors on three counts with its complete range of innovative services:

- Consulting incorporates comprehensive technical advice, from system evaluation through network planning to project planning.
- Training offers you an introduction to the basics, product briefing and user training with certification.
  You find the training courses on technology and products currently available at https://www.belden.com/solutions/customer-innovation-center.
- Support ranges from the first installation through the standby service to maintenance concepts.

With the Customer Innovation Center, you decide against making any compromises in any case. Our client-customized package leaves you free to choose the service components you want to use.

Internet:
https://www.belden.com/solutions/customer-innovation-center

HIRSCHMANN

A BELDEN BRAND