

[English](#) • [Deutsch](#) • [Español](#) • [Français](#) • [Italiano](#) • [???](#) • [Polski](#) • [Português](#) • [??????](#) • [Svenska](#) • [???\(???\)?](#) • [???\(???\)?](#)

Contents

- [1 Introduction \(move to Entware\)](#)
- [2 Introduction \(historic\)](#)
 - ◆ [2.1 Prerequisites](#)
- [3 Example scenario of installing some packages on WRT54GS \(v1.0 .. v3.0\)](#)
 - ◆ [3.1 Creating drive space for your Optware packages](#)
 - ◇ [3.1.1 Option A: Preparing /jffs for /opt](#)
 - ◇ [3.1.2 Option B: Preparing USB drive for /opt](#)
 - ◇ [3.1.3 Option C: Partition-in-a-file on CIFS for /opt](#)
 - ◆ [3.2 Installing ipkg-opt and uclibc-opt](#)
 - ◆ [3.3 /opt backup](#)
 - ◆ [3.4 Search PATH](#)
 - ◆ [3.5 Installing ncFTP client](#)
 - ◆ [3.6 Installing torrent client](#)
 - ◆ [3.7 Installing Throttling HTTP server](#)
 - ◆ [3.8 Installing Samba2](#)
 - ◇ [3.8.1 Setting up smb.conf](#)
 - ◇ [3.8.2 Adding user "nobody" to DD-WRT passwd file](#)
 - ◇ [3.8.3 Setup other users to use the Samba Share](#)
 - ◇ [3.8.4 Check the Samba log files](#)
 - ◇ [3.8.5 Protect your FLASH memory](#)
- [4 Running at startup](#)
 - ◆ [4.1 Startup for external USB Disk](#)
- [5 Known bugs](#)
 - ◆ [5.1 ngrep doesn't work](#)
- [6 Optware on CIFS - Partition in a file](#)
- [7 Remarks](#)

Introduction (move to Entware)

A newer option is available (as of 2015). It is called 'Entware'. For more see: [Installing Entware or Entware Github page](#)

Please note:

- That Brainslayer disabled FPU emulation in changeset 20047 which breaks Optware on DD-WRT.
- BrainSlayer build [19519](#) might be the newest build that is compatible with Optware
- [Kong K2.6 compiled builds](#) are however not affected by this change.

If you still want to use Optware:

Optware

- Also see [Optware, the Right Way](#) for a more updated, user-friendly tutorial, on how to use optware on older DD-WRT builds (until build [19519](#)).
- or continue reading this page....

Introduction (historic)

With the latest DD-WRT (v23 SP2 and v24 SP1) (until build [19519](#)), it is possible to install [Optware packages](#) from [NSLU2-Linux](#) project. This port of Optware is tailored for DD-WRT and extends the router to full feature linux if enough storage ([jffs](#), [SD/MMC Card](#), [USB](#)) is provided for packages.

The main feature of this package is that it installs all required files under /opt with 1000 ported applications. With buildroot toolchain, system library independence is also added. This means that packages now depend on /opt/lib/uClibc.so and dynamic loader also in /opt/lib with separate /opt/etc/ld-opt.so.conf for custom .so cache. Applications are built with wchar and locale support which makes uClibc library somewhat greater than other uClibc packages, but still far smaller than glibc based applications.

Buildroot toolchain features:

- GCC 4.1.1
- uClibc 0.9.28 with many patches
- uClibc++ 0.2.2
- [Native POSIX Threads Library \(NPTL\) Support for uClibc](#)

Some packages (perl, python, Transmission, buildroot, ...) are so memory hungry that swap is needed for normal operation. Buildroot package or optware-devel meta package provides complete development toolchain for compiling programs on router itself with gcc, gdb, make, g++ , svn, ...

Basic preparation of /jffs is required (enable, clean). On WRT54GS there is room for some applications. SD card (/mmc) may give enough space for many applications. For testing purposes also smbmount can be used. If a non /jffs filesystem is available, it is recommended to unmount /jffs and then mount eg. mmc in place of /jffs.

Prerequisites

As always here are three prerequisites.

- Hardware
 1. Recommended are routers with USB support (Asus WL-500g DeluxePremium, Linksys WRTSL54GS).
 2. WRT54g routers with hardware modification for MMC/SD card
 3. 32MB of RAM may not be enough for some application and thus swap is recommended
 4. USB hard disk for storage and swap or USB key without swap
 5. Routers with 8MB flash can also fit some applications in /jffs without the need for swap (e.g. WRT54GS v1.1)
- Software

Optware

1. DD-WRT v23 SP2 mini or standard when MMC is used
2. You should be able to use telnet and the command line.
3. DD-WRT v24 SP1 Mega (or eko special version with built-in USB drivers) when USB is used

- Writable `/opt` folder, which can be mounted on one of the following writeable drives

1. jffs drive
2. USB drive
3. CIFS - Client Samba drive
4. mmc Flash drive

Example scenario of installing some packages on WRT54GS (v1.0 .. v3.0)

Creating drive space for your Optware packages

Prerequisite for optware is writable `/opt` partition/drivespace. Required size depends on packages installed. Three options to create writable partition/drivespace is listed here. A fourth option could be the SDmod.

Option A: Preparing `/jffs` for `/opt`

Depending on the dd-wrt firmware version you choose, you may have enough space on the `/jffs` drive. Example: WRT54GS with 8MB flash and DD-WRT v23SP2 Mini leaves 5248k blocks for `/jffs` which can be binded to `/opt`.

```
~ # mkdir /jffs/opt
~ # mount -o bind /jffs/opt /opt
```

Read more about built-in flash drive option jffs2

Use command *df* to keep track of how much space is left on your `/jffs` and `/opt` "drives"

Option B: Preparing USB drive for `/opt`

Read USB storage wiki for guide to enable, partition, format and mount a USB drive. Note that `/opt` must be mounted on an ext2/3 filesystem for linking to work which is required by the Optware installer.

Option C: Partition-in-a-file on CIFS for `/opt`

For a detailed example of how to create a partition-in-a-file, on a CIFS drive, and mount `/opt` to that partition, goto section Optware on CIFS - Partition in a file below.

Installing ipkg-opt and uclibc-opt

First we will install uClibc library required by all packages and better ipkg-opt packager. Recommended way for installing basic two packages is with installation script that will also check for possible network problems and /opt directory availability. To install (or reinstall) issue the following command:

lamsao: the above links seem to be dead. I found this link and it works.[dead links deleted]

```
wget http://www.3iii.dk/linux/optware/optware-install-ddwrt.sh -O - | tr -d '\r' > /tmp/optware-i
sh /tmp/optware-install.sh
```

Output should look like:

```
Checking system config ...
Using 192.168.1.1 as default gateway.
Using the following nameserver(s):
nameserver 192.168.1.30
Warning: local nameserver is different than gateway!
Check config or enter:
  sed -i s/192.168.*/192.168.1.1/ /tmp/resolv.conf
to correct this.
Installing package uclibc-opt_0.9.28-13_mipsel.ipk ...
Connecting to ipkg.nslu2-linux.org[140.211.166.82]:80
uclibc-opt_0.9.28-12 100% |*****| 832 KB 00:00:00 ET
Updating /opt/etc/ld.so.cache
/opt/sbin/ldconfig: can't create /opt/etc/ld.so.cache~ (No such file or directory)
Installing package ipkg-opt_0.99.163-9_mipsel.ipk ...
Connecting to ipkg.nslu2-linux.org[140.211.166.82]:80
ipkg-opt_0.99.163-9_ 100% |*****| 75896 00:00:00 ET
Downloading http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/Packages.gz
Inflating http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/Packages.gz
Updated list of available packages in /opt/lib/ipkg/lists/optware
Successfully terminated.
Installing uclibc-opt (0.9.28-12) to /opt/...
Downloading http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/uclibc-opt_0.9.28-12_mi
package uclibc-opt suggests installing ipkg-opt
Configuring uclibc-opt
Updating /opt/etc/ld.so.cache
Successfully terminated.
Installing ipkg-opt (0.99.163-9) to /opt/...
Downloading http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/ipkg-opt_0.99.163-9_mip
Configuring ipkg-opt
Successfully terminated.
```

Besides pre-installing fundamental packages, there are also [Optware manual installation instructions](#). This are now somewhat deprecated due to large number of user miss-understandings.

/opt backup

Now we have prepared /opt for adding more packages. We can also create backup of the whole /opt tree in case of router upgrade, to give preinstalled /opt to a friend or just safety backup. We will create such backup with

```
~ # tar czf /tmp/opt-backup.tar.gz /opt
```

Optware

Destination for /tmp/opt-backup.tar.gz should be large enough! Beware that /tmp is RAM. Destination should be disk or remote smbfs partition. Extracting /opt backup could be:

```
cd /
rm -rf opt/*
tar xvzf /tmp/opt-backup.tar.gz
```

If /opt space is tight and no packages requiring libstdc++ will be used (like Amule) then removing this large shared library is recommended

```
~ # rm /opt/lib/libstdc\*
```

will remove

```
/opt/lib/libstdc++.so          /opt/lib/libstdc++.so.6      /opt/lib/libstdc++.so.6.0.8
```

ipkg-opt downloads packages into /opt directory and erases them after installing. If you have enough free RAM and /jffs only then adding --tmp-dir /tmp is recommended instead of default /opt

Search PATH

Note: as of EKO svn11218 (dec 2008) it is no longer necessary to add to path or profile. As the mentioned changes are already added to the default path and profile.

For your scripts it may however still be beneficial to adjust LD_LIBRARY_PATH and PATH, as shown below, and as used in [Startup for external USB Disk](#) example.

Adding /opt/bin to search PATH will save us from tedious typing of /opt/bin for each command from /opt/bin. For running applications from /opt tree search path should be expanded with

```
~# export PATH=/opt/bin:/opt/sbin:$PATH
```

Now we can install any package from a list of available packages.

```
~# ipkg-opt list | more
```

or less or this [Quick list of Optware packages](#)

Note: DD-WRT sets LD_LIBRARY_PATH in /etc/profile that can cause library loading clash. Unset this variable with

```
unset LD_LIBRARY_PATH
```

Installing ncFTP client

We want to get list of packages containing name ftp

```
~ # /opt/bin/ipkg-opt list '*ftp*'
atftp - 0.7-6 - Advanced TFTP server and client
ftpcopy - 0.6.7-3 - A simple FTP client written to copy files or directories (recursively) from a
ftpd-topfield - 0.6.10-2 - FTPD for the Topfield TF5000PVrt
ncftp - 3.2.0-2 - Nice command line FTP client
```

Optware

```
proftpd - 1.3.0a-1 - Highly configurable FTP server with SSL-TLS
tftp-hpa - 0.45-1 - A tftp package
tnftp - 20050625-1 - tnftp (formerly lukemftp) is what many users affectionately call the enhance
tnftpd - 20061217-2 - tnftpd (formerly lukemftpd) is a port of the NetBSD FTP server to other sys
vsftpd - 2.0.1-7 - ftp daemon with an emphasis on speed and security
Successfully terminated.
```

For example we will install ncFTP client with:

```
~ # /opt/bin/ipkg-opt --tmp-dir /tmp install ncftp
Installing ncftp (3.2.0-2) to /opt/...
Downloading http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/ncftp\_3.2.0-2\_mipsel.ipk
Installing ncurses (5.6-1) to /opt/...
Downloading http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/ncurses\_5.6-1\_mipsel.ipk
Configuring ncftp
Configuring ncurses
Successfully terminated.
```

and run it

```
~ # /opt/bin/ncftp ftp.si.debian.org
NcFTP 3.2.0 (Aug 05, 2006) by Mike Gleason (http://www.NcFTP.com/contact/).
```

```
Copyright (c) 1992-2005 by Mike Gleason.
All rights reserved.
```

```
Connecting to 164.8.6.249...
CAMTP Debian mirror [ftp.si.debian.org]
Logging in...
Anonymous access granted, restrictions apply.
Logged in to ftp.si.debian.org.
ncftp / > ls
backports/      debian-cd/      debian-sl/      openoffice/
debian/          debian-non-US/  favicon.ico     robots.txt
ncftp / > bye
```

You have not saved a bookmark for this site.

```
Would you like to save a bookmark to:
ftp://ftp.si.debian.org
```

```
Save? (yes/no) yes
Enter a name for this bookmark, or hit enter for "si": debian
Bookmark "debian" saved.
```

Always check available size of your /opt partition with *df* command

```
~ # df
Filesystem          1k-blocks      Used Available Use% Mounted on
/dev/root              1984           1984          0 100% /
/dev/mtdblock/4       5248           3860          74% /jffs
/dev/mtdblock/4       5248           3860          74% /opt
```

Remove unneeded packages with

```
~ # /opt/bin/ipkg-opt remove ncftp
Removing package ncftp from /opt/...
```

Installing ncFTP client

Optware

Successfully terminated.

You will notice that ncurses as depended package will not get removed. To correct this issue

```
~ # /opt/bin/ipkg-opt remove ncurses
Removing package ncurses from /opt/...
Successfully terminated.
```

unless some other package require this package.

Installing torrent client

Of course there are many more packages which should run on router without a problem. All binaries have built in /opt/lib search path and there should be no problem with libraries clash like problems with OpenWRT packages. All packages have builtin package dependencies and will install required libraries automagically. Lets show this on installing torrent clients enhanced-ctorrent (Stopped developing since June 2008), Enhanced CTorrent is a revised version of CTorrent:

```
~ # /opt/bin/ipkg-opt -t /tmp install enhanced-ctorrent
Installing enhanced-ctorrent (dnh2.2-6) to /opt/...
Downloading http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/enhanced-ctorrent\_dnh2.2
Installing libuclibc (0.2.2-4) to /opt/...
Downloading http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/libuclibc\_0.2.2-4\_mipsel
Installing openssl (0.9.71-3) to /opt/...
Downloading http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/openssl\_0.9.71-3\_mipsel
Configuring enhanced-ctorrent
Configuring libuclibc
Configuring openssl
Successfully terminated.
```

We will not show running example for this clients as normally huge amount of storage is required.

Another torrent client with WWW interface is Transmission. Now it has two branches listed in Optware.

Transmissiond

Lightweight BitTorrent daemon with CGI WWW interface, has a rich-functional web interface watching daemon state and controlling its work, transmission_watchdog can automatically watch and process torrent files in a directory. This one has lower memory footprint, but the development has stopped.

Transmission

Joshe created completely different "daemon" from scratch and maintained as one branch of transmission. Now it has merged the Clutch web interface, and listed as transmission in the list of Optware. The Clutch has better look but less function. The daemon can watch a directory for new torrent files now but you need to config this via it's setting file. Also the current build v1.93-1 (transmission-daemon, transmissison-remote, transmissioncli) seems buggy.

Below is a example of installing transmissiond (the old transmission). For example of installing the one with Clutch web interface please hit [Transmission \(v1.93-1\)](#) Note the config file between the two daemons are not compatible.

Optware

Installing transmissiond (v1.42-1) is simple as

```
~ # /opt/bin/ipkg-opt install transmissiond
Installing transmissiond (1.42-1) to /opt/...
Downloading http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/transmissiond_1.42-1_mip
package transmissiond suggests installing gnuplot
package transmissiond suggests installing logrotate
package transmissiond suggests installing tthttpd
package transmissiond suggests installing mini-sendmail
package transmissiond suggests installing transmission
Configuring transmissiond
Please note that /opt/etc/init.d/S80busybox_httpd is installed for
running CGI interface at http://localhost:8008/cgi-bin/transmission.cgi
See instructions for daemon and WWW interface in /opt/share/doc/transmissiond
Configure daemon with /opt/etc/transmission.conf
Successfully terminated.
```

Here you don't need to install the 'transmission', cause transmissiond is good enough. You can read [/opt/share/doc/transmissiond/README.daemon](#) for instructions. Here are quick steps:

After creating directories with

```
~ # mkdir /opt/torrent
~ # mkdir /opt/torrent/source
~ # mkdir /opt/torrent/work
~ # mkdir /opt/torrent/target
```

we will configure transmissiond itself by editing [/opt/etc/transmission.conf](#) by using

```
~ # sed -i -e 's|/tmp/harddisk|/opt|' /opt/etc/transmission.conf
```

to change default path from [/tmp/harddisk](#) to [/opt](#). So your [transmission.conf](#) will look like

```
~ # cat /opt/etc/transmission.conf
SOURCE=/opt/torrent/source
WORK=/opt/torrent/work
TARGET=/opt/torrent/target
ACTIVE=/opt/torrent/active-torrents.txt
HOME=/opt/torrent
.....
```

Also user should be changed from admin to root. (Change `USER=root` in [/opt/etc/transmission.conf](#))

After changing config `transmission_watchdog` should be run every half an hour with cron. We will start it manually and check if it is running

```
~ # /opt/sbin/transmission_watchdog
~ # ps |grep transmission
10621 root          808 S   transmissiond -p 65534 -w 300 -u -1 -d -1 -i /opt/var/run/transmission.
10624 root          808 S   transmissiond -p 65534 -w 300 -u -1 -d -1 -i /opt/var/run/transmission.
10625 root          808 S   transmissiond -p 65534 -w 300 -u -1 -d -1 -i /opt/var/run/transmission.
```

Transmissiond installation comes with [/opt/etc/init.d/S80busybox_httpd](#) to run its web interface. However, we will configure "Throttling HTTP server" (see next section) to show the startup page at port 8008. Although `dd-wrt v23SP2` uses [/opt/sbin/httpd](#) (`httpd -> /opt/bin/busybox`) for its WEB interface, this httpd server cannot

Optware

be reused as it has serious problems running CGI programs and showing index.html files.

For public trackers CGI interface provides 'Fetch' button which downloads .torrent file into /opt/torrent/source directory. Private trackers use passkey which means that .torrent file should be saved with browser itself.

After seeing .torrent file in Queued list, one can press Watchdog button to force queue processing. This will move .torrent file from source into work directory and notify daemon for the change to start processing the file. After download finishes, transmission_watchdog will automatically move torrent from work to target. Meanwhile, user can press Update button to see status of active torrents.

Besides transmissiond two other daemons are required for normal queue processing. Both are provided by dd-wrt if enabled in the web-GUI. "Cron" is enabled under Admin control panel. "Syslog" is enabled under Services tab.

Cron should regularly run /opt/sbin/transmission_watchdog with sample /opt/etc/crontab

```
*/30 * * * * root /opt/bin/nice /opt/sbin/transmission_watchdog
```

Visit [CRON](#) page to see where this line should be included. I recommend

```
cat /opt/etc/crontab >> /tmp/crontab
```

at startup,

Syslog will collect cumulative reports from transmissiond and CGI will show it when Log or Update is pressed. If gnuplot is installed then also transfer graph is showed when Log is pressed. SYSLOG line in /opt/etc/transmission.conf must be changed to

```
SYSLOG=/opt/var/log/messages
```

to be able to see Log messages from daemon when Log button is pressed.

By default after enabled syslogd in the dd-wrt web-GUI, syslogd logs messages to /tmp/var/log/messages which is in ram. You can create S05syslogd in /opt/etc/init.d/ as below so it will log messages into disk everytime the router is rebooted.

```
#!/bin/sh
# startup script for syslogd
#stop meself if running
if [ -n "`pidof syslogd`" ]; then
    killall syslogd 2>/dev/null
fi
# remove old syslog.log
rm /tmp/var/log/messages
# add symlink so log is available at old location(e.g for webinterface)
ln -s /opt/var/log/messages /tmp/var/log/messages
/sbin/syslogd -m 0 -O /opt/var/log/messages -S -l 7
```

Make it executable by `chmod +x /opt/etc/init.d/S05syslogd`

Opening default listening port 65534 for Transmission is another required task which can be solved by

Optware

DD-WRT control panel or adding

```
/usr/sbin/iptables -I INPUT 1 -p tcp --dport 65534 -j logaccept
```

to startup file.

To test if the listening port has been opened properly and can accept incoming connections: First install the text based browser: lynx

```
ipkg-opt install lynx
```

then issue (assume your port is 65534)

```
lynx http://www.utorrent.com:16000/testport2.php?port=65534
```

This will tell you if your port is OK.

Installing Throttling HTTP server

For our purposes thttpd WEB server will handle most requirements like browsing directories and running CGI program located at `/opt/share/www/cgi-bin/transmission.cgi`

```
~ # /opt/bin/ipkg-opt install thttpd
Installing thttpd (2.25b-4) to /opt/...
Downloading http://ipkg.nslu2-linux.org/feeds/optware/ddwrt/cross/stable/thttpd\_2.25b-5\_mipsel.i
Configuring thttpd
```

To complete the installation, you must edit `/opt/etc/thttpd.conf`, and then run `/opt/etc/init.d/S80thttpd` to start thttpd.

Successfully terminated.

Edit `/opt/etc/thttpd.conf` with vi (or nano) and add `cgipat`, change user to root, change port to 8008 to see the following contents:

```
dir=/opt/share/www
port=8008
user=root
nochroot
nosymlink
novhost
logfile=/var/log/thttpd.log
pidfile=/var/run/thttpd.pid
cgipat=cgi-bin/*
```

Manually start thttpd with

```
~ # /opt/etc/init.d/S80thttpd start
```

After that you will be able to see default index.html page at <http://192.168.1.1:8008/> assuming that your router is at 192.168.1.1 . There is also link to CGI page <http://192.168.1.1:8008/cgi-bin/transmission.cgi> which is pretty empty with several buttons for controlling daemon. Fetch command is useful only for fetching torrent

files without passkey as there is no possibility to pass cookies to server. Use with care. Proper fetch would be "download in browser and then upload to server (accessing server using samba, ftp or scp)".

Installing Samba2

Samba2 the easy way

See [Slim Samba2](#) - a stripped down version of Optware Samba 2 for use on USB drives, just unpack a .tar.gz file, adjust a few settings in the web-GUI, and you are up and running.

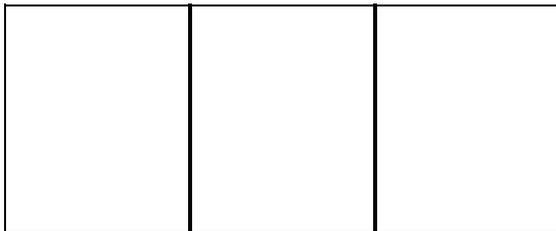
See [HDSamba2](#) - a fully automated install and boot script for installing Optware Samba2 with SWAT, and more... on a USB HD. Format you HD, adjust a few settings in the web-GUI, run the install script, and you are up and running.

Samba2 the step-by-step way

It looks like Samba 3 is too big for all "swapless" routers. Normally Samba 3 will use 10M RAM when up and will use more RAM when new connections established and it forks new threads. Samba 2 will roughly use half of this amount of RAM. For this reason I have prepared Samba 2 packages with Samba Web Admin Tool (SWAT) which will ease config and admin for samba newcomers.

But with Samba 2, you will get a slower data transfer speed (1.5M for Samba 3 and 1.3M/s for Samba 2 over wireless), this is a trade off for the RAM. If you have 64M RAM and have a external hard disk for swap, you can go for Samba 3. Also, you can run Samba as a xinetd service instead of a daemon, so it will only up on request. [Running Samba 2 with xinetd, not as Daemon](#)

SWAT interface looks like:



SWAT globals SWAT shares SWAT status

Installation of samba 2 is done by the following commands:

```
/opt/bin/ipkg-opt update
/opt/bin/ipkg-opt remove samba
/opt/bin/ipkg-opt install samba2
/opt/bin/ipkg-opt install xinetd
```

Note about optware xinetd on dd-wrt: If you get a "can't resolve symbol '__ctype_b_loc'" error, means xinetd doesn't load the right lib from your environment setting. You could try: `/opt/sbin/xinetd -d`. The recommended approach is adding this line at the top of `/opt/etc/init.d/S10xinetd` to override the environment setting for xinetd :

Optware

```
export LD_PRELOAD=/opt/lib/libuClibc-0.9.28.so
```

Now run Xinetd and Samba

```
/opt/etc/init.d/S10xinetd start  
/opt/etc/init.d/S80samba start
```

Point your browser to <http://192.168.1.1:901/> to administer samba. (Here xinetd starts the SWAT when there's a request on port 901)

Installation of xinetd is optional and it is not required if one wants to manually edit `/opt/etc/samba/smb.conf`

Setting up smb.conf

Setting the "correct" (read: usable) settings in the `/opt/etc/samba/smb.conf` file seems to be an art all to itself.

Here is an example that some have found to work well:

```
## Samba config file created using Notepad++  
  
## Global parameters  
[global]  
  ##Change the following three lines as you see fit:  
  server string = Samba Server  
  workgroup = workgroup  
  netbios name = DD-WRT  
  
## without this line nmbd won't start and you won't be able to connect using the netbios name  
  interfaces = 192.168.1.1/255.255.255.0  
  
## The security setting seems important for easy access from Windows  
security = SHARE  
  
## The encrypt password seems important for easy access from Mac OS X  
encrypt passwords = yes  
  
smb passwd file = /opt/etc/samba/smbpasswd  
null passwords = Yes  
# lanman auth = No  
# wins support = Yes  
# name resolve order = wins hosts lmhosts bcast  
  
# Setting code page info, seems to allow more of the european letters  
# in filenames to be correct.  
client code page = 850  
character set = ISO8859-1  
  
## The large sockets are usefull on gigabit ethernet  
# socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192  
  
# The logfile path should be adjusted to a better place, but this works out-of-the-box  
log file = /tmp/sambalog.%m  
#log file = /opt/var/samba/log.*  
  
max log size = 100  
load printers = No  
dns proxy = No  
hosts allow = 192.168. 127.
```

Optware

```
## Remote announce seems important for easy access from Linux
## adjust to the broadcast IP of your LAN
remote announce = 192.168.1.255

[homes]
    comment = Home Directories
    read only = No
    browseable = No

[printers]
    comment = All Printers
    path = /opt/var/spool/samba
    printable = Yes
    browseable = No

    [data]
    comment = Common user data space
    path = /mnt/data
    read only = No
    guest ok = Yes

#[tmp]
#    comment = Temporary file space
#    path = /tmp
#    read only = No
#    guest ok = Yes

#[jffs]
#    path = /jffs
#    read only = No
#    guest ok = Yes

#[mmc]
#    path = /mmc
#    read only = No
#    guest ok = Yes
## The following options can be added to any of the shares if you like
#    force user = root
#    inherit permissions = Yes
#    inherit acls = Yes
#    status = Yes

#[root]
#    path = /
#    read only = No
#    guest ok = Yes
```

For the above smb.conf settings to work, you need to add

```
grep -q nobody /etc/passwd || echo "nobody:*:65534:65534:nobody:/var:/bin/false" >> /etc/passwd
```

to your startup script, or to the top of the `/opt/etc/init.d/S80samba` script

You can read more about smb.conf [here](#) - especially interesting for speed optimizing

You can read more about Code pages in Samba2 here: <http://www.faqs.org/docs/samba/ch11.html>

Optware

If you prefer "security=user" settings, then the following settings in windows might still let you have access to your samba shares

If you are using Vista/Windows 7 Beta, with the stock security settings of the OS, you won't be able to log in into Samba 2's share when your Samba 2 share mode set to USER. The issue is related to an incompatible authentication protocol that runs between Vista/Windows 7 Beta and Samba 2 (lot's of NAS running it). This can be overcome by editing the security settings in windows. See also [Instructions for Using Windows Vista with a Network File Server \(NAS\)](#) Also, Samba 3 doesn't have this issue.

1. Goto Start > Run
2. Type "gpedit.msc" and push enter.
3. Goto Computer Configuration / Windows Settings / Security Settings / Local Policies / Security Options.
4. Find Network Security: LAN Manager authentication level
5. Change settings from: *SendNTLMv2 response only* to: *Send LM & NTLM - use NTLMv2 session security if negotiated*

Or you can edit it directly using regedit:

1. start | run | regedit
2. Find the key KEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa
3. Set LmCompatibilityLevel to 1

Adding user "nobody" to DD-WRT passwd file

DD-WRT does not include user nobody at startup (user nobody is the Linux default for "guest")

This can be overcome with adding this at startup with */opt/etc/init.d/S80samba* script like:

```
#!/bin/sh
[ -d /tmp/samba ] || mkdir /tmp/samba
grep -q nobody /etc/passwd || echo "nobody:*:65534:65534:nobody:/var:/bin/false" >> /etc/passwd
killall smbd
killall nmbd
/opt/sbin/nmbd -D
/opt/sbin/smbd -D
```

Setup other users to use the Samba Share

Adding users to samba *The username need to be already in your /tmp/etc/passwd. And need to use the tool /opt/bin/smbpasswd to add that user into the Samba password file which by default is /opt/etc/samba/smbpasswd (configured in smb.conf). For example, add user root will be:*

```
/opt/bin/smbpasswd -a root
```

Check the Samba log files

Try to look at the log files created by Samba, and check what happens.

Optware

- Default location of log files are: `/opt/var/samba/log.*`

use command: `ls /opt/var/samba/log.*` - to see names of all log files

use command: `cat /opt/var/log/samba/log.nmbd` - to have one log file displayed

use command: `cat /opt/var/log/samba/log.nmbd | more` - to have one log file displayed, one page at a time

use command: `vi /opt/var/log/samba/log.nmbd` - to use vi editor be able to move cursor up and down through the log file.

Type: `:q!` to exit vi editor

- Other location of log files can be specified in `smb.conf`

in section `[global]`

use line: `log file = /tmp/sambalog.%m`

- Default location of `smb.conf` file is: `/opt/etc/samba/smb.conf`

use command: `cat /opt/etc/samba/smb.conf | grep 'log file'` - to see if special log path is specified in `smb.conf`

Protect your FLASH memory

A long term problem that could happen when running samba on FLASH memory like jffs, mmc or USB stick is FLASH wear. Such use could destroy FLASH if log files and browsing databases are constantly updated.

Use of `/tmp` partition is recommended for such use. Add to startup script (`/opt/etc/init.d/S80samba`):

```
echo "Starting smbd:"
if [ ! -d /tmp/samba ]; then
    mkdir /tmp/samba
    mkdir /tmp/samba/log
    rm -rf /opt/var/log/samba
    ln -s /tmp/samba/log /opt/var/log/samba
    rm -rf /opt/var/samba
    ln -s /tmp/samba /opt/var/samba
fi
```

or

if you have the line `[-d /tmp/samba] || mkdir /tmp/samba` in your samba start script, then simply add `log file = /tmp/samba/sambalog.%m` to `smb.conf` under `[global]`, like:

```
[global]
log file = /tmp/samba/sambalog.%m
```

Running at startup

With dd-wrt.V24 the difference between system and OpenWRT Whiterussian R6 uClibc is so big, that running even simple programs causes segmentation fault. Optware also uses its own uClibc that resides in `/opt/lib` but all programs have built-in rpath to `/opt/lib`. This means that it is simpler to use Optware packages under DD-WRT than OpenWRT packages.

Optware

DD-WRT provides user startup files in `/jffs/etc/config` . To create optware startup file do

```
~ # mkdir /jffs/etc
~ # mkdir /jffs/etc/config
~ # vi /jffs/etc/config/optware.startup
```

with the following contents

```
#!/bin/sh

mount -o bind /jffs/opt /opt

unset LD_LIBRARY_PATH
unset LD_PRELOAD

[ -e /opt/etc/profile ] && mount -o bind /opt/etc/profile /etc/profile

if [ -d /opt/etc/init.d ]; then
  for f in /opt/etc/init.d/S* ; do
    [ -x $f ] && $f start
  done
fi
```

and mark it as executable with

```
~ # chmod +x /jffs/etc/config/optware.startup
```

After reboot this file will be executed. I recommed first trying it manually

```
~ # /jffs/etc/config/optware.startup
```

Creating `/opt/etc/profile` with the following contents

```
export PATH=/opt/bin:/opt/sbin:/bin:/sbin:/usr/bin:/usr/sbin
export PS1='\u@\h:\w\$ '

[ -x /opt/bin/less ] || alias less=more
[ -x /opt/bin/vim ] || alias vim=vi

arp() { cat /proc/net/arp; }
ldd() { LD_TRACE_LOADED_OBJECTS=1 $*; }
reboot() { killall nmbd
           killall smbd
           killall xinetd
           killall tthttpd
           sleep 1
           umount /opt
           /sbin/reboot
           }
```

is also recommended as this will unset `LD_LIBRARY_PATH` provided by stock `/etc/profile`

Although ext3 filesystem uses journaling that prevents major corruption and faster recover during crash, it is recommended that one writes its shutdown scrip as alias for `/etc/profile` that kills all optware daemons, umounts filesystems and then issue `/sbin/reboot`. This is what we call *clean shutdown*.

Optware

Note: as of EKO svn11218 (dec 2008) it is no longer necessary to add to path or profile. As the mentioned changes are already added to the default path and profile. For your scripts it may however still be beneficial to adjust LD_LIBRARY_PATH and PATH, as [Startup for external USB Disk](#) example shows below.

For more Startup options with dd-wrt see: [Startup Scripts](#) and [Script Execution](#)

Startup for external USB Disk

The following script is recommended when using external USB hard disk. The main feature of this script is prevention of disk corruption and automatic replacement.

It also shows how LD_LIBRARY_PATH can be used to compensate uClibc incompatibility between dd-wrt /lib and OpenWRT /jffs/lib .

The following script relies on two OpenWRT packages just to provide filesystem checking and enabling swap at startup. Besides required USB kernel modules, two disk related packages were installed: e2fsprogs and swap-utils. All others packages could be used from Optware.

```
#!/bin/sh
# /jffs/etc/config/optware.startup script for USB disk support
# Recommended ext3 partitions:
# /dev/discs/disc0/part1 for /opt 1GB
# /dev/discs/disc0/part2 for swap 512MB
# /dev/discs/disc0/part3 /mmc for user files aka MultiMediaCenter
# Tune partition checks to max 10 counts disregarding last check time
# tune2fs -i 0 -c 10 /dev/discs/disc0/part1
# tune2fs -i 0 -c 10 /dev/discs/disc0/part3
# Required OpenWrt packages: /bin/ipkg install swap-utils e2fsprogs
(
unset LD_PRELOAD
export PATH=/bin:/usr/bin:/sbin:/usr/sbin
export LD_LIBRARY_PATH=/lib:/usr/lib

echo "Inserting modules for USB disk support..."
insmod /jffs/lib/modules/2.4.30/usbcore.o
insmod /jffs/lib/modules/2.4.30/ehci-hcd.o
insmod /jffs/lib/modules/2.4.30/scsi_mod.o
insmod /jffs/lib/modules/2.4.30/usb-storage.o
insmod /jffs/lib/modules/2.4.30/sd_mod.o
insmod /jffs/lib/modules/2.4.30/jbd.o
insmod /jffs/lib/modules/2.4.30/ext3.o

echo "Waiting for modules to initialize disk access..."
sleep 20

# assure correct paths when running OpenWRT programs like e2fsck
export LD_LIBRARY_PATH=/jffs/lib:/jffs/usr/lib
export PATH=/jffs/bin:/jffs/sbin:/jffs/usr/sbin:/jffs/usr/bin:/bin:/sbin

echo "Activating swap..."
swapon /dev/discs/disc0/part2

echo "Checking optware partition..."
e2fsck -p /dev/discs/disc0/part1
```

Optware

```
echo "Checking user-data partition"
e2fsck -p /dev/discs/disc0/part3

echo "Mountint disk partitions..."
export LD_LIBRARY_PATH=/lib:/usr/lib
mount -t ext3 -o noatime /dev/discs/disc0/part1 /opt
mount -t ext3 -o noatime /dev/discs/disc0/part3 /mmc

if [ -e /opt/etc/profile ]; then
    cp /opt/etc/profile /tmp
    mount -o bind /tmp/profile /etc/profile
fi

# provide Optware search paths
unset LD_LIBRARY_PATH
export PATH=/opt/bin:/opt/sbin:/bin:/sbin:/usr/sbin:/usr/bin

echo "Starting Optware programs..."
if [ -d /opt/etc/init.d ]; then
    for f in /opt/etc/init.d/S* ; do
        [ -x $f ] && $f start
    done
fi
) > /tmp/optware.log 2>&1
```

Forcing filesystem recheck on next boot could be marked with

```
tune2fs -C 100 /dev/discs/disc0/part1
```

Known bugs

ngrep doesn't work

ngrep is compiled with IPV6 support and ddwrt doesn't provide IPV6 support.

If ngrep is recompiled without IPV6 support, it works ok after adding the nobody user to /etc/passwd

```
# grep nobody /etc/passwd
nobody:x:99:99:Nobody:/:
```

Optware on CIFS - Partition in a file

[added by --[omegadraconis](#) 02 August 2010

I just took a note from the jffs mount and decided to use mount bind:

```
mount -o bind /cifs1/Router /opt
```

I am sure there are reasons I should not do it this way but, I couldn't find any and I didn't have to load a bunch of kmods.

Optware

[added by --[gruad23](#) 27 June 2010]

I prefer using a cifs-share itself as /opt and install optware as to the way Azuretech describes using a loop-mounted ext2-container on the cifs-share. I described the way I did it on [[knowwiki:optware_on_cifs](#)]

[added by --[Azuretech](#) 19:14, 20 February 2009 (CET)]

I have successfully used a SMB share for /opt, well, actually used a CIFS share, but the technique is the same, regardless. The trick is to create an ext2 partition inside of a file on the CIFS/SMB share, and mount that using the loop device. I have this working on v24. The technique is as follows:

Update: I discovered since I went to a mega build that this no longer works, because of lack of JFFS on some mega builds. Not to worry, here is a fix for this little problem. First disable JFFS in the GUI on DD-WRT, since it doesn't work anyways, and you can't mount it elsewhere if its active. After mounting a CIFS share, simply create a subdirectory in the CIFS share called jffs then:

```
mount --bind /tmp/smbshare/jffs /jffs
nvramp set sys_enable_jffs2=1
mkdir -p /jffs/tmp/ipkg
ipkg update
```

First, you must have a successfully mounted SMB or CIFS share.

After that, you'll have to install two necessary packages. This is done in two ways, depending on your kernel version.

K2.4

If you are running **K2.4**, you should use the ipkg way.

```
ipkg install kmod-loop
ipkg install kmod-ext2
```

ipkg should insert the modules automatically, but just to be sure execute the following commands:

```
insmod /jffs/lib/modules/2.4.30/loop.o
insmod /jffs/lib/modules/2.4.30/ext2.o
```

K2.6

Users running **Eko K2.6** have to install the modules manually using the files below (the kernel module extension starting from v2.6 is *.ko*).

- [Ext2 \(ext2.ko\)](#)
- [Loop \(loop.ko\)](#)

Updated version of loop.ko (working in build 15943)

- [Loop \(loop.ko\)](#)

The modules should be loaded like this:

```
insmod /jffs/lib/modules/2.6.24.111/loop.ko
insmod /jffs/lib/modules/2.6.24.111/ext2.ko
```

After loading the modules, create a container file on your cifs share, in my case I'm creating a 10MB file below.

```
dd if=/dev/zero of=/tmp/smbshare/opt.ext2 bs=1 count=1 seek=10M
```

Now we need to format the empty file as an ext2 file system

```
echo y | mke2fs -L optware /tmp/smbshare/opt.ext2
```

Now that the file is formatted, all we need to do is mount the file:

```
mount -o loop /tmp/smbshare/opt.ext2 /opt
```

then get the optware install script:

```
wget http://www.3iii.dk/linux/optware/optware-install-ddwrt.sh -O - | tr -d '\r' > /tmp/optware-i
```

and install it:

```
sh /tmp/optware-install.sh
```

And there you have it, optware is installed onto your CIFS or SMB directory.

Now that this part is complete, so long as you don't wipe your /jffs directory, you won't need to re-complete this every time you boot, since optware is stored safely on your CIFS file system, I simply added the following to a startup script in my CIFS share that will execute once the CIFS share is mounted.

Copy the appropriate lines for your kernel version.

K2.4

```
insmod /jffs/lib/modules/2.4.30/loop.o
insmod /jffs/lib/modules/2.4.30/ext2.o
mount -o loop /tmp/smbshare/opt.ext2 /opt
```

K2.6

```
insmod /jffs/lib/modules/2.6.24.111/loop.ko
insmod /jffs/lib/modules/2.6.24.111/ext2.ko
mount -o loop /tmp/smbshare/opt.ext2 /opt
```

And now that you can run all those cool optware apps, you're going to want a swapfile so you don't run out of memory.... This is a bit tricky... You can just create a swap file on the SMB share, swapon will protest, you need to create and mount an ext2 file on the SMB share, and then create the swapfile on that. Here's the sequence I followed on the BS mega builds:

Optware

```
ipkg install swap-utils
dd if=/dev/zero of=/tmp/smbshare/swap.ext2 bs=1 count=1 seek=64M
echo y | mke2fs -L swap /tmp/smbshare/swap.ext2
mkdir /tmp/swap
mount -o loop /tmp/smbshare/swap.ext2 /tmp/swap
dd if=/dev/zero of=/tmp/swap/swapfile bs=1024 count=65536    notice this is a bit different 1024
mkswap /tmp/swap/swapfile
chmod og=rw /tmp/swap/swapfile    not sure if this is absolutely necessary, but just to be safe
swapon -pl /tmp/swap/swapfile
```

Once this is done, you just need to add to the end of that CIFS startup script above the following:

```
mkdir /tmp/swap
mount -o loop /tmp/smbshare/swap.ext2 /tmp/swap
/jffs/usr/sbin/swapon -pl /tmp/swap/swapfile
```

So now your swap file will automatically be enabled shortly after boot up. To make sure your swap file is working just issue `swapon -s` you should get something like this back:

```
Filename Type Size Used Priority
/tmp/swap/swapfile file 65536 0 1
```

UPDATE: Now, I have discovered that apparently, some builds are not compiled with swap support, and so even though you install the swap-utils package, you get a response of "can't resolve symbol 'swapon'" when you try to activate the swap file with the swapon command. If this should happen to you, you can resolve this by instead of installing the swap-utils package, install the optware busybox package with `/opt/bin/ipkg-opt install busybox` then calling swapon as `/opt/sbin/swapon /tmp/swap/swapfile` So, the correct sequence to follow in this case is:

First, make sure you've followed the directions above to install Optware, then:

```
/opt/ipkg-opt install busybox
dd if=/dev/zero of=/tmp/smbshare/swap.ext2 bs=1 count=1 seek=64M
echo y | mke2fs -L swap /tmp/smbshare/swap.ext2
mkdir /tmp/swap
mount -o loop /tmp/smbshare/swap.ext2 /tmp/swap
dd if=/dev/zero of=/tmp/swap/swapfile bs=1024 count=65536
/opt/sbin/mkswap /tmp/swap/swapfile
chmod og=rw /tmp/swap/swapfile
/opt/sbin/swapon /tmp/swap/swapfile
```

Special thanks to whoever the contributor of the information at <http://forum.x-wrt.org/index.php?topic=428.0:wap2> is, as it was incredibly helpful in figuring out how to accomplish this.

--[Azuretech](#) 06:49, 29 August 2009 (CEST)

For some reason I had problems creating the **swap.ext2** and **swapfile** files using the commands above on my router. Instead, I created them on another computer running Linux and then copied them to the router.

On the Linux PC:

```
dd if=/dev/zero of=swap.ext2 bs=1024 count=8192 (8MB swap partition)
scp swap.ext2 root@dd-wrt:/tmp/smbshare
```

Optware

Mount it on the router:

```
echo y | mke2fs -L swap /tmp/smbshare/swap.ext2
mkdir /tmp/swap
mount -o loop /tmp/smbshare/swap.ext2 /tmp/swap
df (note the reported size of /tmp/swap)
```

Create the swapfile on the Linux computer:

```
dd if=/dev/zero of=swapfile bs=1024 count=7509 (no larger than what df showed!)
scp swapfile root@dd-wrt:/tmp/swap
```

Finally, enable swap on the router:

```
/opt/sbin/mkswap /tmp/swap/swapfile
chmod og=rw /tmp/swap/swapfile
/opt/sbin/swapon /tmp/swap/swapfile
free (verify swap space is now available)
```

Also see [Linux SWAP](#).

--[Disk Crasher](#) 02:44, 29 October 2009 (CET)

Remarks

- DD-WRT provided /bin/ipkg is not full feature ipkg but it can do the job for installing ipkg-opt. It could also be used for installing other packages, but this is not recommended any more.
- DD-WRT kernel lacks many "server" features like NFS, NBD support, ... Provided samba, mysql, thttp-php and other applications can do the job if enough space is provided. This means that external storage is required for memory hungry applications. Experience from nslu2 and Oleg firmware showed that 512MB swap space is recommended if external USB hard disk is available. See [USB](#) page for instructions how to initialize USB kernel modules and harddisk.
- Do not use VFAT filesystem for /opt. Use ext2 or ext3 instead. Alternatively one can use loop device to mount filesystem in file that resides on VFAT. See <http://www.nslu2-linux.orghttp://wiki.dd-wrt.com/wiki/FAQ/Optware-uClibcBuild> for example on doing this on USB stick.
- Optware for DD-WRT is beta-stage feature (as of 2006). Report problems and suggestions in [DD-WRT forum](#) under [Optware](#).
- For instructions on some other packages look at <http://www.nslu2-linux.orghttp://wiki.dd-wrt.com/wiki/Unslung/Packages>
- Optware [package tracking](#) and repository access
- There is also possibility to install Optware packages under Oleg, X-wrt or OpenWRT Whiterussian firmware. See instructions on <http://www.nslu2-linux.orghttp://wiki.dd-wrt.com/wiki/FAQ/Optware-uClibcBuild>

[Oleo](#)