

Contents

- [1 USB storage](#)
 - ◆ [1.1 Applications](#)
 - ◆ [1.2 Compatible units](#)
- [2 DD-WRT versions with built-in USB support](#)
 - ◆ [2.1 Step 1 - Partition and format your USB drive](#)
 - ◆ [2.2 Step 2 - Connect the USB drive to your DD-WRT box](#)
 - ◆ [2.3 Step 3 - Enable USB support on the Mega / Mini USB version of DD-WRT](#)
 - ◇ [2.3.1 Prevent Harddisk Spindown](#)
 - ◆ [2.4 Step 4 - Enable Network Access](#)
 - ◇ [2.4.1 Network Option A - Use SSH \(SCP\) and WinSCP](#)
 - [2.4.1.1 General intro to the dd-wrt folder structure](#)
 - ◇ [2.4.2 Network Option B - Use NAS \(FTP\)](#)
 - [2.4.2.1 FTP Option - allow WAN FTP access](#)
 - [2.4.2.2 FTP Option - allow anonymous FTP access to all or part of the USB drive](#)
 - ◇ [2.4.3 Network Option C - Use Samba and Windows Network Neighbourhood](#)
 - [2.4.3.1 Samba2 the easy way](#)
 - [2.4.3.2 Updated Instructions](#)
 - ◇ [2.4.4 Network Option D - Use NFS](#)
 - ◆ [2.5 USB Guide for v24 and v24SP1 Mega of DD-WRT](#)
 - ◇ [2.5.1 V24-Step 3.A - Prepare to mount the drive\(s\)](#)
 - ◇ [2.5.2 V24-Step 3.B - Mount the drive/partitions](#)
 - ◇ [2.5.3 V24-Step 3.C - Auto mount partition at mount-time / boot-time](#)
 - [2.5.3.1 Option 3C.1 - Auto-mount using web-GUI](#)
 - [2.5.3.2 Option 3C.2 - Alternative Auto mount at boot-time](#)
 - ◇ [2.5.4 Step 6 - Downloading extra Software Packages](#)
 - [2.5.4.1 Option 6.1 - Extra software the OpenWRT way](#)
 - [2.5.4.2 Option 6.2 - Extra software the Optware way](#)
 - ◇ [2.5.5 V24-Step 7 - Auto-run the extra software at boot-time](#)
 - ◆ [2.6 To access a networked USB printer](#)
- [3 DD-WRT versions WITHOUT built-in USB support](#)
 - ◆ [3.1 Prerequisites](#)
 - ◆ [3.2 Install USB Kernel modules on jffs](#)
 - ◆ [3.3 Manual load and check of modules](#)
 - ◇ [3.3.1 Check](#)
 - ◆ [3.4 Create a startup script that loads the modules](#)
 - ◇ [3.4.1 For DD-WRT v23 SP1 or below](#)
 - ◆ [3.5 kernel modules for filesystems](#)
 - ◇ [3.5.1 Load filesystem modules](#)
 - ◆ [3.6 Multislot card reader](#)

- [4 Usage](#)
 - ◆ [4.1 Mounting](#)
 - ◆ [4.2 Performance Testing](#)
 - ◇ [4.2.1 On a WRTSL54G with 4G flash disk](#)
 - ◇ [4.2.2 On a WRT350N with 2GB SanDisk Cruzer Titanium flash disk](#)
 - ◇ [4.2.3 On a Buffalo WBMR-HP-G300H with WD MyPassport 500GB](#)
 - ◆ [4.3 Listing Contents](#)
 - ◆ [4.4 If you're having trouble](#)
- [5 Links](#)

USB storage

Applications

- The use of a [USB-compatible router](#) makes the [SD/MMC modification](#) unnecessary. Just drop any old digital camera card into a USB flash card reader and easily expand the storage space of your Linux-based router. Any of the various card types, including CompactFlash and SD, are suitable. USB flash devices are also available as one self-contained unit: a small "tag" which can store up to several gigabytes of data.
- A USB hard drive can be used as network-attached storage, shared by all computers on a LAN using SSH (scp), FTP, [Samba](#) or [NFS](#). Most standard internal hard drives (notebook or desktop) can be converted simply by installing them in an inexpensive USB enclosure.
- A USB hard drive can also be used as virtual memory by creating swap space. Instead of abruptly running out of memory when the 32Mb of physical RAM in these units is full, a portion is swapped to disc and the applications continue to run normally, but with a speed penalty.
- Mounting a USB storage device as /opt provides nearly-unlimited space to install many applications from the large collection of [Optware](#) packages, ported to embedded Linux originally for use with the Linksys "slug" (NSLU2) USB network-attached storage servers. A USB-compatible router can provide pretty much all of the NSLU2's capabilities (at a comparable price) plus offer all of the capabilities of a standard router and a wireless access point.

The use of standard USB hubs allows other USB devices (such as printers, serial ports, CD/DVD drives and audio) to remain connected even while the USB interface is used for flash cards or hard drive interfaces.

Compatible units

This is not a comprehensive list of [Supported Devices](#), just a few which support USB:

- [ASUS WL-500g Premium](#) - [USB on ASUS WL-500gP.v2](#)
- [Asus WL-500W](#) - specs. [1]
 - ◆ DD-WRT v24 sp1 will brick this router! v.24sp2 [2] NEWD/NEWD-2 mini and mega will work fine. If using JFFS2 do **NOT** flash with BrainSlayer-V24-preSP2 builds!!
- [Asus WL-520GU](#) - specs. [3]
- [Asus RT-N16](#)
- [Belkin Play Max F7D4301v1](#) - specs. [4]

USB_storage

- ◆ do not enable USB 1.1 ohci - it will crash at mom. tested with v24 sp2 15030 Newd-2 K26.
- [Buffalo WZR-HP-G300NH](#)
- [Linksys WRT600N](#)
- [Linksys WRT610N](#) Versions 1 and 2
- [D-Link DIR-320](#) - specs. [5]
- [D-Link DIR-330](#)
- [D-Link DIR-632](#)
- [Linksys WRTSL54GS](#) - specs. [6]
- [Linksys WRT600N](#) - specs. [7]
- [Linksys WRT350N](#) - specs. [8]
- [TP-Link TL-WR1043ND](#)

Note: The Asus WL-520GU requires the use of the usb-ohci driver. Using the ehci-hcd driver will identify and find USB storage devices, but reboots the WRT every time you try to access the storage. See [this OpenWRT forum discussion](#) for more information. These issues seem to have been resolved on the newest v24_TNG versions - See this forum thread: [USB on BCM5354 routers - need testers Fixed as of V.24sp2](#)

DD-WRT versions with built-in USB support

In most cases, if the router has a USB port and supports DD-WRT. Then the DD-WRT firmware built for that router will have USB support.

- As of v.24 USB support is included via the web-GUI in the Mega versions of dd-wrt.

Note: v.24 x86 version of dd-wrt does not yet have USB support built in to the web-GUI

- Eko has expanded the USB options of DD-WRT v24_TNG versions (TNG = The Next Generation = V24-preSP2): <http://www.dd-wrt.com/dd-wrtv3/dd-wrt/downloads.html>

(under 'others', 'Eko', 'v24_TNG', from svn10137 or later)

(from version v24_TNG-svn11218 USB and NAS (ftp) seems to be working smoothly, both in the new Mega (for 8Mb Flash routers) and new Mini_usb_ftp (for 4Mb flash Routers))

(Early v24_TNG versions only works with one USB drive. As I was testing svn11536 I noticed that several USB drives were now automatically detected correctly. Though the plug-in of a second drive did seem to confuse the "Run-on-mount" settings - your mileage may vary).

- At one point there were issues on new routers using the BCM5354 CPU, these issues seem to have been resolved on the newest v24_TNG versions - See this forum thread [USB on BCM5354 routers - need testers](#)
- User lightix has prepared a proof of concept [micro_plus_usb_asus520gc.bin](#) version that combined with [mega.tar.gz](#) unpacked on a USB drive, gives the option to run DD-WRT Mega on routers with as little as 2Mb FlashRAM and a USB stick. Read more about testing this concept at this [English forum](#) and this [Russian forum](#)

Step 1 - Partition and format your USB drive

NOTE - The following information is very outdated and pretty much useless for K26 capable routers. It is recommended that you read [Optware, the Right Way](#) before you continue!

See [How to - Format and Partition External Storage Device](#) for more detailed instructions.

You need to first partition and format your drive

This can be done by attaching the drives to an other system.

Like a PC running linux (maybe from a LiveCD) or from a Windows PC using a partitioning tool.

Some have been able to use their iPod.

You can format your partitions to either ext2/3 or FAT32.

Use ONLY ext2/3 for partition/drives that you plan for permanent connection to your DD-WRT box, *or* for drives where you plan to install additional software packages onto (like optware, Samba, Torrent etc.). You can use any of the three mount points: /opt, /mnt, /jffs

Use FAT32 on data partitions if you intend to connect this drive to other systems (MAC, Windows) to copy files from/to directly. You should use the mount point: /mnt

For USB flash drives it is recommended to use ext2 or FAT32. The journaling system of ext3 file system on USB flash drives will add to write-wear of the flash chip.

Note: for EKO v24TNG-svn11218 Mini_usb_ftp - does not support ext2 formatted partitions.

For large HD drives that you "permanently" attach to your router you may want to consider creating three or four partitions:

- one: for optware packages - make it 32MB - 2GB - use ext2/3 as the format
- two: for swap file - make it 16-256 MB - format it as linux swap file
- three: for data space - make it fit the rest of the disk - use ext2/3 or FAT32 as the format
- four: for jffs space - make it 32MB - 2GB - use ext2/3 as the format

Note: With the third partition being the data space, your HD drive will work with the SlimSamba2 and HDsamba2 scripts described below.

Step 2 - Connect the USB drive to your DD-WRT box

Plug in the freshly partitioned and formatted USB drive to you dd-wrt box!

Step 3 - Enable USB support on the Mega / Mini_USB version of DD-WRT

Using the Web-GUI to enable USB support select:

USB_storage

Note: as of version Eko [V24TNG-svn11205](#) USB and NAS (ftp) has their own sub-tabs under the Services tab

Services -> Services --> USB support:
Core USB support = **enable**

Enable one or both of the following if you want to use USB 1.1 devices

- USB 1.1 Support (UHCI)
- USB 1.1 Support (OHCI)

Note: The Asus WL-520GU requires the use of the usb-ohci driver. Using the ehci-hcd driver will identify and find USB storage devices, but reboots the WRT every time you try to access the storage. See this [OpenWRT forum discussion](#) for more information. **Fixed as of v.24sp2**

Enable the following if you want to use USB 2.0 devices

- USB 2.0 Support (EHCI)

Enable the following if you want to access USB hard disks or USB flash drives

- USB Storage Support = **enable**

Activate one or more file systems to be able to access on the storage devices

- ext2 / ext3 File System Support
- FAT File System Support

NOTE: not all dd-wrt versions have embedded filesystems support (Although these options are accessible via WEB-interface). If your hardware has flash 4MB or less - it may not include filesystems kernel modules. Try to search your filesystem for files ext2.ko, ext3.ko, vfat.ko and so on. If they are absent, your system doesn't support filesystems and you need some kind of trick

[Mounting USB drive without located onboard fs modules.](#)

Enable the following if you want to access USB printers attached to the DD-WRT mega/usb loaded router over the network

- USB Printer Support

As of EKO build [V24TNG-svn11100](#) the following options are also available:

Note: as of version [V24TNG-svn11205](#) USB and NAS (ftp) has their own sub-tabs under the Services tab

Prevent Harddisk Spindown

This is a common issue and is easily mitigated by the following commands (used through an Ubuntu Linux Distro). **Replace the ? with appropriate drive letter**

USB_storage

Read **THIS** first <http://www.dd-wrt.com/phpBB2/viewtopic.php?t=57799>

Example Only:

```
sudo apt-get install sdparm
sudo sdparm -a /dev/sd?
```

You'll get an output like this:

```
/dev/sd?: Seagate FreeAgent Pro 400A
Power condition mode page:
IDLE 0 [cha: n, def: 0, sav: 0]
STANDBY 1 [cha: y, def: 1, sav: 1]
ICT 0 [cha: n, def: 0, sav: 0]
SCT 9000 [cha: y, def:9000, sav:9000]
```

You want to change STANDBY to 0 (this will also make SCT 0, I believe the 9000 is how many seconds until it spinsdown)

Run the following commands:

```
sudo sdparm --command=start /dev/sd?
sudo sdparm --clear STANDBY -6 /dev/sd?
sudo sdparm -a /dev/sd?
```

The output now looks like this:

```
/dev/sd?: Seagate FreeAgent Pro 400A
Power condition mode page:
IDLE 0 [cha: n, def: 0, sav: 0]
STANDBY 0 [cha: n, def: 1, sav: 0]
ICT 0 [cha: n, def: 0, sav: 0]
SCT 0 [cha: n, def:9000, sav: 0]
```

Copied and modified from <http://ubuntuforums.org/showthread.php?t=679228>

Edit: You can use something like /dev/scsi/host0/bus0/target0/lun0/disc for device name on dd-wrt. To revert STANDBY changes use this:

```
sudo sdparm --defaults --page=po -6 /dev/sd?
```

Step 4 - Enable Network Access

After activation of USB storage support, you need to select one or more ways to access the USB drive space over the network. You have several options, and you can select one or more of these options.

Using SSH (and WinSCP) has been an option for a long time with dd-wrt.

As of EKO [V24TNG-svn11218](#) dd-wrt also has a built-in FTP server option.

This guide will also show some options of using Samba and NFS to enable Network Access to the drivespace of your DD-WRT system.

Note: V24_TNG-svnxxxxx are pre-SP2 development versions of dd-wrt. For the most part they work just fine - and they have the new USB and NAS functionality. However: they are still development versions. Generally

USB_storage

speaking, if you need the new USB and NAS functionality, you should install the version with the highest recommended svn number. See [Peacock Forum Thread \(Note 3.\)](#) for current recommendations.

Network Option A - Use SSH (SCP) and WinSCP

One advantage of using SSH and SCP protocol to access your files is that passwords and data are encrypted during network transfer.

In the web-GUI select:

tab: Services -> tab: Services -> section: Secure Shell -> option: SSHd = enable
Click "Save" and "Apply Settings"

You can now access the file system using the SCP protocol.

To open for WAN access to your files using the web-GUI select:

tab: Administration -> tab: Management -> section: Remote Access -> option: SSH Management = enable
Click "Save" and "Apply Settings"

File manager programs for using SSH (scp)

- [WinSCP](#) is an open source file manager for windows.

WinSCP quick user guide: Hostname would be the URL or IP number of your router. User name is you "root" and "root-password". Select the SCP protocol. Ignore the two warnings about group when first accessing your router.

You will start in the /tmp/root folder, move to the /mnt, /opt or /jffs folder (as set on your USB tab in the web-GUI), to access your USB drive.

General intro to the dd-wrt folder structure

All folders in the root (/) is basically read-only, until you mount one of the folders on a read-write-able (rw) enabled drive.

◇ /tmp is mounted on the RAM drive of dd-wrt.

Special folders are /jffs, /opt, /mnt and /mmc. These folders have special dd-wrt uses, and the special functionality and special hardware can be activated using the web-GUI. Due to the nature of Linux any folder can also be mounted/re-mounted on any other drive/sub-folder in the system, but to use these options you should know what you are doing.

◇ /jffs is usually mounted on the "left-over" built-in flash space, and is usually activated in the web-GUI. See [Jffs](#)

◇ /mmc is usually mounted on SD Flash Ram cards of a [SD/MMC-mod](#), and is usually activated in the web-GUI.

◇ /opt is usually mounted anywhere on a writeable drive with enough space, and is mainly needed for use with the [Optware](#) package system

◇ /mnt can be used for any rw drive, and does not yet have any specific functionality in the

USB_storage

web-GUI connected with it.

Linux/dd-wrt "commands" can be found in /bin, /sbin, /usr/bin and /usr/sbin

You can also read more about using WinSCP with DD-WRT on the [WinSCP](#) wiki page.

Network Option B - Use NAS (FTP)

As of version Eko [V24TNG-svn11218](#) dd-wrt Mega and Mini_USB has a built-in FTP server option.

In the web-GUI select:

tab: Services -> tab: NAS -> section: ProFTPD ->

option: ProFTPD = enable

option: Server Port - 21 - is recommended

option: Files Directory - /mnt - is recommended, but should just be the same as set in the USB tab.

option: Allow Write - enable - is needed if you want to be able to FTP-write to the drive

option: User Password List -> any *name* [space] *password* can be entered

Name and password must be separated by a space.

You should enter at least one *user name* and *password*, so you do not need to use the "root-user" to get FTP access.

Click "Save" and "Apply Settings"

And you should now have FTP LAN access to your USB drive

FTP Option - allow WAN FTP access

To allow WAN access using FTP protocol run the following line in a terminal window:

```
/usr/sbin/iptables -I INPUT 1 -p tcp --dport 21 -j logaccept
```

Note: FTP is a clear text protocol, so your FTP username and password can be sniffed, so you should think twice before enabling WAN FTP access.

FTP Option - allow anonymous FTP access to all or part of the USB drive

You can allow anyone on your LAN to read all or a specific folder-tree on your USB drive. If you enable WAN access, the anonymous settings will also apply to all WAN users (= all of the internet users)

option: Anonymous Login (Read-only) - can be enabled if you want anyone to be able to read files on your USB drive.

option: Anonymous Home Sub-directory - can be set to a sub-folder where you keep your public files and folders, like: /mnt/public (in Anonymous Home Sub-directory text field write: /public)

Anonymous example - you must create the public folder in a terminal or using the normal FTP user.

```
cd /mnt
mkdir public
```

USB_storage

In Anonymous Home Sub-directory text field write: **/public**
Click "Save" and "Apply Settings"

Network Option C - Use Samba and Windows Network Neighbourhood

Samba is the Linux tool for sharing drives with Windows (and MACs and Linux) over the network using the SMB protocol (= Windows Network Neighbourhood).

- Check out the [Optware](#) tutorial for standard Samba options.
- [forums.slickdeals.net](#) also has a very extensive write-up of installing Optware Samba and using USB drives.
- See [Links section](#) for even more examples.

Samba2 the easy way

- [Slim Samba2](#) - A quick, easy and slim version of Optware Samba2.
 - ◇ For easy use on a USB flash drive (or USB HD), it can be installed using the web-GUI and just four CLI commands.
 - ◇ You may even unpack the Slim Samba2 files on the USB drive, while it's still attached to the Linux system you use to format the drive with. Then adjust settings in the web-GUI, plug-in the USB drive to your router, and you are up and running.
 - ◇ If you are new to dd-wrt or Linux this may be a good place for you to start.Slim Samba2 can be used with any USB enabled firmware newer than Dec. 17, 2008 (svn 11218).
- [HDSamba2](#) is an easy to use install and boot script system that will:
 - ◇ check your setup,
 - ◇ download needed OpenWRT and Optware packages,
 - ◇ install and trim settings so your USB drive, dd-wrt, jffs drive and Samba2 with SWAT will all
 - ◇ be up and running in a matter of minutes.
 - ◇ HDSamba2 boot script helps you maintain pristine data integrity of your USB drive.HDSamba2 can be used as is with any USB enabled firmware newer than Dec. 17, 2008 (svn 11218). Some users have also used it with V24SP1 firmwares.

Updated Instructions

I found a great guide here, but even it's not entirely up to date with what I have in one of the latest builds (r25760). So here is a set of simplified instructions to help another newb (like me!) get started.

To enable USB: - go to Services > USB - Enable "Core USB Support"; this will open up some new options - Enable "USB Storage Support" - Enable "Automatic Drive Mount" - Enable "Use SES Button to remove drives" (optional) - click Save

Plug in your USB storage device. Wait for it to appear in the Disk Info area. Note the mount path (likely

USB_storage

/tmp/mnt/sda1).

To enable access to your USB device on your Windows network: - go to Services > NAS - Enable "Samba" - enter your Windows "Workgroup" name - under "File Sharing > Shares" click Add Share - select the "Path" (likely /mnt/sda1) - give your USB device a "Name" - under "File Sharing > Users" click Add User - enter a "User Name", "Password", and select the name of the device you added above under "Access Shares" and then "samba" - click Apply Settings

Open Windows Explorer and go to Network, where you should now see a "dd-wrt" entry. When you click it, you will be prompted to enter the User Name and Password you created in Services > NAS above. Click OK and you should have access to your USB storage device. You could either map this as a network drive (right-click and select "Map network drive"), or add it to your Favorites (right-click Favorites at the top of the Windows Explorer tree and select "Add current location to Favorites").

Not a command line entry or package to compile in the whole process--amazing! Big thanks again to 'rvhamcontrolviation' for setting me off along the right path! Hope this helps someone else...

Network Option D - Use NFS

see [NFS tutorial](#)

(to do: update / expand the NFS tutorial)

USB Guide for v24 and v24SP1 Mega of DD-WRT

As development of DD-WRT happens ALL the TIME, it is somewhat difficult to keep the Wiki fully up-to-date, as it basically has to cover both V23, V24, V24SP1 and V24TNG. And for USB and NAS significant development (and change) has happened during these versions.

"Old" information is still kept here, as it has some instructional/educational value, for those who want/need to create their own tweaks to the DD-WRT USB setup.

V24-Step 3.A - Prepare to mount the drive(s)

- for manual mounting of partitions, you need to identify mounting details about you USB drive(s)
- Use a telnet connection (use ie. [PuTTY](#)) to connect to your router
- Use command **dmesg** to get mounting details of your USB drive(s). The list can get quite long, so you need to...

Look for something that starts with /dev, like:

Partition check:

/dev/scsi/host0/bus0/target0/lun0: p1

p1 means partition1.

USB_storage

The string `/dev/scsi/host0/bus0/target0/lun0` is hardware specific, which is why you need to find this information/string for *your* specific router. The example is from a wl500gP v1 router.

Command `ls` can then be used to find actual name of partition(s) on your drive

Example with three partitions:

```
root@DD-WRT:~# ls /dev/scsi/host0/bus0/target0/lun0
disc  part1  part2  part3
root@DD-WRT:~# ls -l /dev/scsi/host0/bus0/target0/lun0
brw----- 1 root    root      8,    0 Jan  1  1970 disc
brw----- 1 root    root      8,    1 Jan  1  1970 part1
brw----- 1 root    root      8,    2 Jan  1  1970 part2
brw----- 1 root    root      8,    3 Jan  1  1970 part3
root@DD-WRT:~#
```

Actual name of partition 3 would be:

`/dev/scsi/host0/bus0/target0/lun0/part3`

Examples of mount commands can also be found at:

this forum post: [how to access USB drive from WL500gP w/ DD-WRT v24 mega?](#)
and at [Usage](#)

V24-Step 3.B - Mount the drive/partitions

You need to mount the drive/partition, using the information you found using the `dmesg` command, and add a `/part1` at the end of the string.

For easy use with Optware packages (where you can easily get ie. Samba, FTP and more), it is recommended that you mount your new partition on existing folder `/opt`

For a HD this command can be used to mount the partition (ex. for wl500gP v1)

```
mount /dev/scsi/host0/bus0/target0/lun0/part1 /opt
```

For a USB flash drive this command can be used to mount the partition

```
mount -o noatime /dev/scsi/host0/bus0/target0/lun0/part1 /opt
```

The '-o noatime' option means: "Do not update inode access times on this file system".
This may cause less writing to the flash drive, thus lengthening its life span.

USB_storage

To mount a fat partition you need to use special mount options to make it writable. Along these lines:

```
mount -o rw,umask=0000 /dev/discs/disc0/part3 /mnt/data
```

See example in this [forum thread](#)

For compatibility with Slim Samba2 or HDsamba2, you might also choose to mount partition1 on existing folder `/mnt`, and then create a `/mnt/opt` folder and "combine" `/mnt/opt` to existing folder `/opt`, this would then be the steps:

```
mount /dev/scsi/host0/bus0/target0/lun0/part1 /mnt
mkdir /mnt/opt
mount /mnt/opt /opt
```

For drives with three partitions the Data Partition can be mounted on any existing folder like `/mmc`, `/tmp/usb1` or `/mnt/data`

Example 1:

```
# data partition (part3) is mounted on existing folder /mmc
mount /dev/scsi/host0/bus0/target0/lun0/part3 /mmc
```

Example 2:

```
mkdir /tmp/usb1
mount /dev/scsi/host0/bus0/target0/lun0/part1 /tmp/usb1
```

Example 3: (which is compatible with Slim Samba2 and HDsamba2)

```
# data partition (part3) is mounted on the folder you made on /mnt/data
mkdir /mnt/data
mount /dev/scsi/host0/bus0/target0/lun0/part3 /mnt/data
```

It is possible to specify the preferred file system for a partition using the `-t` option. After `-t` specify the file system you would like to use. The following example mounts `/mmc` as an ext3 file system

```
mount -t ext3 /dev/scsi/host0/bus0/target0/lun0/part3 /mmc
```

Use command `ls /lib/modules/2.4.36/` to see which file systems are available on your dd-wrt version. Look for `ext2.o`, `ext3.o`, `vfat.o`, `fat.o` and others (with v.24)

Note: If you plan to do much testing of installing software, you may consider mounting the first drive as `/mnt`, and then making test sub-directories/folders for `/opt` mounts, as this will give you a quick way of switching the entire `/opt` folder system from one to another test setup (see [Option 3C.2](#) for example).

V24-Step 3.C - Auto mount partition at mount-time / boot-time

As of EKO 11218 (V24preSP2) it is possible to auto-mount the first partition on the USB drive using settings in the web-GUI.

If you want the drive/partition to be auto mounted on V24SP1 at boot-time, or if you want to do special things at boot-time (any version), you need to:

option a) enable jffs (a user accessible partition on the built-in flash EEPROM of the router) , and store a startup script in the `/jffs/etc/config` folder.

In this folder, scripts with extension: `.startup` will automatically be run when your dd-wrt router boots.

option b) or save the the command to mount a partition, and run a boot-script file in NVRAM

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

To use option a) look at the [jffs: Directions for using Web-GUI](#) to enable the jffs partition. Then store `usb.startup` script in the `/jffs/etc/config` folder

After enabling the jffs partition (and rebooting your router) to download the script you need to : Write the following commands in a telnet/ssh session:

```
mkdir /jffs/etc
mkdir /jffs/etc/config
cd /jffs/etc/config
wget http://www.3iii.dk/linux/dd-wrt/usb.startup
chmod +x usb.startup
```

Example of the `usb.startup` script

```
#!/bin/sh

# Mount partition 1 as /opt folder - then wait for the drive to mount
mount /dev/scsi/host0/bus0/target0/lun0/part1 /opt
sleep 5

# set path so optware binaries are automatically found as new commands
echo "export PATH=$PATH:/opt/bin:/opt/sbin" > /tmp/root/.profile
export PATH=$PATH:/opt/bin:/opt/sbin

# option for auto-starting standard Samba2 Optware install
/opt/etc/init.d/S80samba start

# method for auto-running scripts found in /opt/etc/config folder
for I in `bin/ls /opt/etc/config/*.startup`
do
    sh $I &
done
```

HDsamba2 uses this *option a)* to run its boot script. HDsamba2 boot script is `/jffs/etc/config/init.startup`

USB_storage

To use **option b)** look at the [Startup Scripts](#) page

Short version: in web_GUI goto tab:Administration -> tab:Commands -> type your commands in the text window -> click:Save Startup

You need to add the *mount* command to the NVRAM stored startup script, only add "&" to the end of the command. You can mount the */opt* folder on a w1500gP v1 with the following command:

```
mount /dev/scsi/host0/bus0/target0/lun0/part1 /opt
```

See also: [Option 3C.2](#) for further example of using this NVRAM boot script method.

Option 3C.1 - Auto-mount using web-GUI

With these settings in the web-GUI of builds newer than Eko [V24TNG-svn11218](#)

- Automatic Drive Mount - select: enable
- Run-on-mount Script - type: */opt/usb.run*
- Disk Mount Point - select: */opt*

The first partition on the USB drive will be mounted as */opt*, and the script */opt/usb.run* can be used to start Samba, FTP, NFS and more.

Type *ls /opt* to see if your drive/partition has been recognised:

```
root@DD-WRT:~# ls /opt
lost+found
root@DD-WRT:~#
```

Slim Samba2 uses this 3C.1 method. Only the settings are:

- Automatic Drive Mount - select: enable
- Run-on-mount Script - type: */mnt/samba2.usb*
- Disk Mount Point - select: */mnt*

Option 3C.2 - Alternative Auto mount at boot-time

Save in NVRAM using web-GUI -> Administration -> Commands

```
mount /dev/scsi/host0/bus0/target0/lun0/part1 /mnt
sleep 3
sh /mnt/etc/config/begin-here.startup
```

Click: Save Startup

Make the following folders:

```
mkdir /mnt/etc
mkdir /mnt/etc/config
mkdir /mnt/test1
mkdir /mnt/main-opt
mkdir /mnt/org-opt
```

Save the following script to the `/mnt/etc/config` folder as file `begin-here.startup`

```
#!/bin/sh

# Mount /opt to one of the options - then wait for the drive to mount
mount /mnt/test1 /opt
#mount /mnt/org-opt /opt
#mount /mnt/main-opt /opt
sleep 3
```

Make the script run-able/executable by the boot process using `chmod` command

```
chmod +x /mnt/etc/config/begin-here.startup
```

Note: V24-Step 4-5 are changed to step V24-Step 3A-C and - Thus we jump to Step 6.

Step 6 - Downloading extra Software Packages

Now you need to activate Samba, FTP or NFS systems to be able to share your USB mounted drive/partition over the network.

These software packages are not pre-installed on DD-WRT, so you need to get them from either OpenWRT or Optware.

option a) You can download packages from OpenWRT using the `ipkg` commands, these packages are directly designed for the OpenWRT system, but many will run on a DD-WRT box, and all will run with minor modifications, as DD-WRT is made to be as compatible with OpenWRT as possible.

◇ [Option 6.1](#) has an example of installing and using an OpenWRT package to test your USB drive data integrity.

option b) Once the initial Optware files are setup you can download the Optware packages using the `ipkg-opt` commands. Optware has cross-compiled almost all of their 1000+ packages specifically to fit DD-WRT needs. Optware also uses the `ipkg` system to download and setup their packages (NSLUG/Optware invented the `ipkg` system!), but for installing Optware on DD-WRT it is recommended to use the DD-WRT - Optware adapted command `ipkg-opt`.

◇ [Option 6.2](#) goes through the steps of installing the basic Optware files onto your USB drive.

USB_storage

- ◇ ipkg-opt note: once the initial optware files are installed, then using "ipkg-opt" or "/opt/bin/ipkg" ensures the correct optware result. If you just use "ipkg" you also have to look at your PATH to know which of these two files ("/bin/ipkg" or "/opt/bin/ipkg") are actually used.
Use command "set" to see the active PATH (or command "echo \$PATH" - PATH must be in capital letters).
- ◇ See more at [Ipkg tutorial](#) and [Optware tutorial](#)

Option 6.1 - Extra software the OpenWRT way

If you have enabled jffs you may want to add some HD tools, so you can check the quality of your HD file system before mounting them. The following command will install standard ext2/3 maintenance tools and the tool needed to use a swap partition.

```
cd /jffs
/bin/ipkg update
/bin/ipkg install e2fsprogs swap-utils
```

note: You'll need abt. 700Kb of jffs free-space to install the e2fsprogs tool/package. Use command "df" to see how much space you have on your drives.

When the e2fsprogs have finished installing, then the following command will make sure that your partition is okay, before you mount it.

```
/jffs/sbin/e2fsck -p /dev/scsi/host0/bus0/target0/lun0/part3
```

note: the -p parameter will automatically fix errors if any are found.

Option 6.2 - Extra software the Optware way

Once you have mounted an /opt folder, then either of the following ways, will install the basic Optware files

```
wget http://pastebin.ca/raw/1031954 -O - | tr -d '\r' > /tmp/optware-install.sh
```

or

```
wget http://www.wlan-sat.com/boleo/optware/optware-install-ddwrt.sh -O - | tr -d '\r' > /tmp/optw
```

or

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

then run: *sh /tmp/optware-install.sh*

Note about installing the basic Optware files: If your are still having trouble getting the script to run, then

USB_storage

you might chance it and edit the `/tmp/optware-install.sh` script, by adding a `#` in front of the `_check_config` line at the bottom of the script. Should then be like this:

```
<snip>
#_check_config
_install_package uclibc-opt_0.9.28-13_mipsel.ipk
_install_package ipkg-opt_0.99.163-10_mipsel.ipk
</snip>
```

Then move to `/tmp` folder, and issue command

```
sh optware-install.sh
```

Note about running Optware on V24 and V24SP1 firmware: To have Optware packages co-exist on dd-wrt, you may need to create a new `/etc/profile` setup. (as of version V24preSP2/v24_TNG: EKO svn11152 and BS 111208 the default `/etc/profile` already has some of these changes built-in and you do not need to make your own profile additions)

Create `/opt/etc/profile` file with the following content

```
export PATH=/opt/bin:/opt/sbin:/opt/usr/bin:/opt/usr/sbin:/bin:/sbin:/usr/bin:/usr/sbin:/jffs/sbi
export LD_LIBRARY_PATH="/opt/lib:/opt/usr/lib:/lib:/usr/lib:/jffs/lib:/jffs/usr/lib:/jffs/usr/loc
export PS1='\u@\h:\w\$ '

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

Now mount the new profile

```
mount --bind /opt/etc/profile /etc/profile
```

You should also modify the `begin-here.startup` file (mentioned in [Option 3C.2](#)) to look like this:

```
#!/bin/sh

# Mount /opt to one of the options - then wait for the drive to mount
mount /mnt/test1 /opt
#mount /mnt/org-opt /opt
#mount /mnt/main-opt /opt
sleep 3

#Update stock profile to Optware optimized profile
[ -e /opt/etc/profile ] && mount -o bind /opt/etc/profile /etc/profile
```

Read more at the [optware](#) wiki entry to see which software packages you can also install, like Samba, FTP, Torrent client, extra HTTP server and more.

V24-Step 7 - Auto-run the extra software at boot-time

Add some way of auto-running optware software like Samba at boot-up time

method a) Start your extra software directly from the NVRAM stored commands (see [Startup Scripts](#)), or better yet

method b) Start your different systems from one script that calls another script, and just call ONE initial script from the NVRAM stored command, see [Option 3C.1](#)

method c) activate jffs2 and place you startup scrits in /jffs/etc/config, as scripts in this folder automatically are run when your router is booted (see: [jffs](#) of how to enable jffs).

To access a networked USB printer

check [USB Printer sharing](#)

2.1 you need to make sure it is recognized by dd-wrt

2.2 You may need to adjust start-up scripts so the printer is correctly initialized

2.3 You need to install the correct printer driver on you pc's OS

2.4 You need to connect the printer driver of your pc to the networked usb printer

DD-WRT versions WITHOUT built-in USB support

If you plan to use USB it is highly recommended that you choose one of the built-in USB support versions and follow the sections above. However, if you really want/need to add USB support by hand, follow the steps bellow.

Prerequisites

To install the USB Kernel modules, a couple of initial steps are required (this assumes a new install starting with DD-WRT default settings).

1. [Enable JFFS](#)

- You will need ~650KB of free storage space in JFFS -- if you don't have enough free space, consider the mini or micro versions of DD-WRT.

2. Ensure the ipkg temporary directory is created (otherwise the update command will fail)

- Telnet/SSH to the WRT

```
# mkdir -p /jffs/tmp/ipkg
```

3. Update ipkg package list

```
# cd /jffs
# ipkg update
```

Install USB Kernel modules on jffs

If your kernel is still 2.4.30 (older versions of dd-wrt -> see output of *uname -a*):

```
ipkg install kmod-usb-uhci // this is for USB 1.1 (only required if your router does not support
ipkg install kmod-usb-core
ipkg install kmod-usb2
ipkg install kmod-usb-storage
```

If your kernel is newer than 2.4.30:

```
ipkg -force-depends install kmod-usb-core kmod-usb2 kmod-usb-storage
```

This has been tested with kernel 2.4.35 and it did work

For the Asus WL-520GU, you will also need the kmod-usb-ohci package:

```
ipkg -force-depends install kmod-usb-ohci
```

Working with a DIR-320 I found some errors which I haven't captured, around decompressing and installing these packages. I was using the mini version that leaves around 444kb of free jffs space. The error was around the control.tar.gz file which wasn't been properly processed, thus, failing the packages to be installed.

After searching a considerable amount of time I found a nice trick that did it. Instead of creating the tmp dir inside /jffs as explained in [jffs#Add USB Storage](#), you can use the standard /tmp which will use the ram instead of the jffs storage. This might be useful for temp files and also because you will probably have more ram than jffs space to work with.

To do this, you have to first remove the tmp dir from your /jffs directory if you created it before by doing:

```
rmdir /jffs/tmp (it has to be empty)
```

Then do:

```
mkdir /tmp/tmpipkg
ln -s /tmp/tmpipkg /jffs/tmp
```

Then try to install packages as explained above. Be aware that each time you reboot the router, the /tmp is cleared so the symbolic link will point to an inexistant directory, which you would have to recreate again. I guess you could just use /tmp for the symbolic link, but this is not something I've tried.

Manual load and check of modules

- load the modules one by one with the following commands in a telnet session.

```
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
```

- for the ASUS WL-520GU, use the usb-ohci driver instead of ehci-hcd (**Fixed as of V24sp2**)

```
insmod /jffs/lib/modules/2.4.30/usbcore.o
insmod /jffs/lib/modules/2.4.30/usb-ohci.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
```

Once you have everything working, you will need to add the insmod commands to your boot script.

Check

- See the enabling from **dmesg** output of a Linksys WRTSL54GS. A SanDisk Cruzer Micro 4G flash drive as recognized.

```
# dmesg
<snip>
<6>ehci_hcd 01:02.2: PCI device 1033:00e0
<6>ehci_hcd 01:02.2: irq 2, pci mem c0181000
<6>usb.c: new USB bus registered, assigned bus number 1
<6>ehci_hcd 01:02.2: USB 2.0 enabled, EHCI 1.00, driver 2003-Dec-29/2.4
<6>hub.c: USB hub found
<6>hub.c: 5 ports detected
<6>hub.c: new USB device 01:02.2-1, assigned address 2
<4>usb.c: USB device 2 (vend/prod 0x781/0x5406) is not claimed by any active driver.
<6>SCSI subsystem driver Revision: 1.00
<6>Initializing USB Mass Storage driver...
<6>usb.c: registered new driver usb-storage
<6>scsi0 : SCSI emulation for USB Mass Storage devices
<0> Vendor: SanDisk Model: U3 Cruzer Micro Rev: 2.18
<0> Type: Direct-Access ANSI SCSI revision: 02
<0> Vendor: SanDisk Model: U3 Cruzer Micro Rev: 2.18
<0> Type: CD-ROM ANSI SCSI revision: 02
<7>WARNING: USB Mass Storage data integrity not assured
<7>USB Mass Storage device found at 2
<6>USB Mass Storage support registered.
<0>Attached scsi removable disk sda at scsi0, channel 0, id 0, lun 0
<0>SCSI device sda: 8015505 512-byte hdwr sectors (4104 MB)
<0>sda: Write Protect is off
<6>Partition check:
<6> /dev/scsi/host0/bus0/target0/lun0: p1 p2 p3
#
```

Create a startup script that loads the modules

Create a file, usb.startup in /jffs/etc/config/ directory

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

that contains this (in vi editor press i once for insert mode):

```
#!/bin/sh
insmod /jffs/lib/modules/2.4.30/usbcore.o
insmod /jffs/lib/modules/2.4.30/ehci-hcd.o
# WL-520GU users - add a dash (#) to the line above, and remove the dash (#) from the line below
#insmod /jffs/lib/modules/2.4.30/usb-ohci.o
insmod /jffs/lib/modules/2.4.30/scsi_mod.o
insmod /jffs/lib/modules/2.4.30/usb-storage.o
# remove the dash (#) from the line below if you use USB 1.1
#insmod /jffs/lib/modules/2.4.30/uhci.o
insmod /jffs/lib/modules/2.4.30/sd_mod.o
```

vi is ended using "ESC" and then :wq

make the script executable:

```
chmod +x /jffs/etc/config/usb.startup
```

For DD-WRT v23 SP1 or below

add the following script to startup. It executes on every reboot all *.startup scripts located in /jffs/etc/config/

console mode

```
> ~ # nvram set rc_startup="
> for I in ` /bin/ls /jffs/etc/config/*.startup`
> do
>   sh $I &
> done
> "
> ~ # nvram commit
>
```

web interface mode

```
for I in ` /bin/ls /jffs/etc/config/*.startup`
do
sh $I &
done
```

This is it. Here's my dmesg from Asus WL-500gP

```
dmesg
SCSI subsystem driver Revision: 1.00
Initializing USB Mass Storage driver...
```

USB_storage

```
usb.c: registered new driver usb-storage
USB Mass Storage support registered.
uhci.c: USB Universal Host Controller Interface driver v1.1
PCI: Enabling device 01:03.0 (0000 -> 0001)
uhci.c: USB UHCI at I/O 0x100, IRQ 2
usb.c: new USB bus registered, assigned bus number 2
hub.c: USB hub found
hub.c: 2 ports detected
PCI: Enabling device 01:03.1 (0000 -> 0001)
uhci.c: USB UHCI at I/O 0x120, IRQ 2
usb.c: new USB bus registered, assigned bus number 3
hub.c: USB hub found
hub.c: 2 ports detected
```

kernel modules for filesystems

Now to access FAT/EXT/XFS partitions you must install one or more of these modules:

```
ipkg install kmod-vfat
ipkg install kmod-ext2
ipkg install kmod-ext3
ipkg install kmod-xfs
```

Example of installing kmod-vfat to /jffs

```
/mnt # ipkg install kmod-vfat
ipkg_depends: Warning: kernel mentioned in dependency but no package found in //jffs/usr/lib/ipkg
Downloading http://downloads.openwrt.org/whiterussian/packages/kmod-vfat_2.4.30-brcm-5_mipsel.ipk
Connecting to downloads.openwrt.org[195.56.146.238]:80
kmod-vfat_2.4.30-brc 100% |*****| 32018    --:--:-- ETA
Done.
ipkg_depends: Warning: kernel mentioned in dependency but no package found in //jffs/usr/lib/ipkg
Unpacking kmod-vfat...Done.
Configuring kmod-vfat...Done.
/mnt #
```

Check what got installed.

```
/mnt # ipkg files kmod-vfat
/jffs
/jffs/lib
/jffs/lib/modules
/jffs/lib/modules/2.4.30
/jffs/lib/modules/2.4.30/vfat.o
/jffs/lib/modules/2.4.30/fat.o
/jffs/etc
/jffs/etc/modules.d
/jffs/etc/modules.d/30-vfat
/mnt #
```

Load filesystem modules

Once you have the needed modules stored on /jffs, then you need to load them for dd-wrt/Linux to use them. This is done with the *insmod* command

Example of /jffs/etc/config/usb.startup boot script which loads usb and filesystem modules. (for your specific use comment out the lines you don't need to load, in order to save memory on your running system)

This first section loads usb modules.

```
/mnt # cat /jffs/etc/config/usb.startup
#!/bin/sh
insmod /jffs/lib/modules/2.4.30/usbcore.o
insmod /jffs/lib/modules/2.4.30/ehci-hcd.o
# WL-520GU users - add a dash (#) to the line above, and remove the dash (#) from the line below
#insmod /jffs/lib/modules/2.4.30/usb-ohci.o
insmod /jffs/lib/modules/2.4.30/scsi_mod.o
insmod /jffs/lib/modules/2.4.30/usb-storage.o
# remove the dash (#) from the line below if you use USB 1.1
#insmod /jffs/lib/modules/2.4.30/uhci.o
insmod /jffs/lib/modules/2.4.30/sd_mod.o
/mnt #
```

The second section of /jffs/etc/config/usb.startup loads the filesystem modules

```
# Filesystems
# Ext2
insmod /jffs/lib/modules/2.4.30/ext2.o
# Ext3
insmod /jffs/lib/modules/2.4.30/jbd.o
insmod /jffs/lib/modules/2.4.30/ext3.o
# FAT
insmod /jffs/lib/modules/2.4.30/fat.o
insmod /jffs/lib/modules/2.4.30/vfat.o
# XFS
#insmod /jffs/lib/modules/2.4.30/xf.o
```

Now *dmesg* should show (The example below is from a 80 GB HD)

```
Attached scsi disk sda at scsi0, channel 0, id 0, lun 0
SCSI device sda: 156301488 512-byte hdwr sectors (80026 MB)
Partition check:
/dev/scsi/host0/bus0/target0/lun0
```

Next step is to mount the drive/partition, see below for steps to do that.

Multislot card reader

In order to use some multislot card readers, you must add the following parameter to the *scsi_mod* in the file /jffs/etc/modules.d/60-usb-storage:

```
scsi_mod max_scsi_luns=8
```

USB_storage

Note that this is model-dependent; some multislots readers do treat each slot as an independent drive and even allow them to be used simultaneously.

Usage

Mounting

Use command `dmesg` to see the specifics of the first parameter for your router

```
mount /dev/scsi/host0/bus0/target0/lun0/part1 /mnt
```

You can also add the above line to `/jffs/etc/config/usb.startup` to mount the partition on reboot

Notes:

- cn0rris: I had to add a `sleep 1` before the mount to get it to mount reliably.
- WL-520GU Users: If you use this script, make sure to replace `ehci-hcd.o` with `usb-ohci.o`
- Manuel: I use `usb-ohci.o` for the `wrt350n`.

An example `/jffs/etc/config/usb.startup` file for use on a device with [JFFS](#), [Optware](#) and all USB storage modules already installed:

```
#!/bin/sh
#
#Load various modules for USB, hard drive, printer/scanner support - omit any of these that aren't
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/fat.o
insmod /jffs/lib/modules/2.4.30/vfat.o
insmod /jffs/lib/modules/2.4.30/ext2.o
insmod /jffs/lib/modules/2.4.30/ext3.o
insmod /jffs/lib/modules/2.4.30/printer.o
insmod /jffs/lib/modules/2.4.30/sg.o
insmod /jffs/lib/modules/2.4.30/scanner.o
sleep 2
#
# Use first partition on first USB hard drive for Optware package storage
mount /dev/discs/disc0/part1 /opt
unset PATH
export PATH=/sbin:/bin:/usr/sbin:/usr/bin:/opt/bin:/opt/sbin:/jffs/sbin:/jffs/bin:/jffs/usr/sbin:
unset LD_LIBRARY_PATH
#
# Enable if using network servers such as SANE or e-mail services
# xinetd
#
# Enable these if using Samba, a Windows-compatible networking package, to share disc drives or C
# nmbd
# smbd
```

USB_storage

```
#
# To enable local network access to the USB-attached printer(s), use one (and only one) of the fo
# cupsd
# /jffs/usr/sbin/p910nd -b -f /dev/usb/lp0 0
#
# Note that these paths appear not to be exported, so you will want to add /opt/bin and /opt/sbin
PATH=/opt/bin:/opt/sbin:/bin:/sbin:/usr/bin:/usr/sbin:/jffs/sbin:/jffs/bin:/jffs/usr/sbin:/jffs
export LD_LIBRARY_PATH=/opt/lib:/opt/usr/lib:/lib:/usr/lib:/jffs/lib:/jffs/usr/lib:/jffs/usr/loca
#
# Enable this *only* if a swapfile has already been created on a USB hard drive and prepared for
# swapon /opt/var/swapfile
#
# Mixing packages from Optware and other sources may lead to conflicts between libraries, so unse
unset LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

The disc drives will also appear as `/dev/discs/disc0/part1` and other corresponding numbers in this series (a symbolic link). Drive 0 nominally appears as the first disc located on startup, typically the drive on the first port of a USB hub. USB card readers may appear as multiple discs if they support multiple card types.

The `.../part1` through `.../part4` are individual primary partitions on a drive, which may be formatted as any of VFAT (DOS/Windows), ext2 or ext3 (Linux native) or Linux swap. It is most often easier to partition and format a USB drive on a desktop PC (preferably a computer with Linux installed, as Windows is limited to creating 32Gb VFAT partitions) before moving it to your embedded device.

Note by Manuel: I had problems with `dd-wrt` detecting only one partition of a three ext2 partition disk and then refusing to mount. After using `gparted` to set the "partition table type" (or the "disk label") in "device" menu to `ms-dos`, the problem went away and the three partitions in the disk were recognized and mounted without problems. I also tried this with the `hfsplus` filesystem but could only mount read-only on `dd-wrt`.

Performance Testing

With help of the `hdparm` package you could test your performance. On an Asus WL-500gP(remium) without overclocking you will get

```
hdparm -tT /dev/scsi/host0/bus0/target0/lun0/part2

Timing cached reads:   188 MB in  2.04 seconds =  92.16 MB/sec
Timing buffered disk reads:  24 MB in  3.19 seconds =   7.52 MB/sec
```

Please note overclocking wouldn't get you significant more performance on Asus WL-500gP(remium) Router.

On a WRTSL54G with 4G flash disk

```
/ # hdparm -tT /dev/scsi/host0/bus0/target0/lun0/part2

/dev/scsi/host0/bus0/target0/lun0/part2:
Timing cached reads:   148 MB in  2.03 seconds =  72.91 MB/sec
Timing buffered disk reads:  24 MB in  3.22 seconds =   7.45 MB/sec
```

USB_storage

```
/ #
```

On a WRT350N with 2GB SanDisk Cruzer Titanium flash disk

```
/ # hdparm -tT /dev/scsi/host0/bus0/target0/lun0/part1

/dev/scsi/host0/bus0/target0/lun0/part1:
Timing cached reads:   196 MB in  2.01 seconds =  97.51 MB/sec
Timing buffered disk reads:  28 MB in  3.04 seconds =   9.21 MB/sec

/ #
```

On a Buffalo WBMR-HP-G300H with WD MyPassport 500GB

```
/ # hdparm -tT /dev/sda1

/dev/sda1:
Timing buffer-cache reads:   86 MB in 0.51 seconds = 169689 kB/s
Timing buffered disk reads:  77 MB in 3.00 seconds = 26254 kB/s
```

Listing Contents

```
ls -l /mnt

# multiple partitions
mkdir /tmp/usb
mkdir /tmp/usb/1
# wait to recognize devices properly
sleep 2
# mount first partition
/bin/mount /dev/scsi/host0/bus0/target0/lun0/part1 /tmp/usb/1
# list its contents
ls -l /tmp/usb/1
# second partition
mkdir /tmp/usb/2
mount /dev/scsi/host0/bus0/target0/lun0/part2 /tmp/usb/2
# list its contents
ls -l /tmp/usb/2
```

If you're having trouble

If you still cannot get your USB drive to mount after following these instructions to the letter, you can try resetting the NVRAM. Do this by using the [recovery utility](#) to upload the bin/trx file meant for that purpose.

NOTE: This has miraculously solved problems on an ASUS WL-500G Premium. Your mileage may vary, especially if you have a different router.

Links

- forums.slickdeals.net - Asus WL-520gU - Example of running DD-WRT with USB (nov 2008)
- blog.davidbarrett.net - Linksys WRT350n - DD-WRT w/ USB, Samba & uPNP (nov 2008)
- [Forum: Official USB support?](#) dd-wrt-forum where key usb development discussion happens (2007-2008)

(please do not ask for install help on this forum thread)
- [Forum: Adding USB support into firmware](#) (vinter 2007-2008) forum discussion on adding the USB drivers to the firmware - v.24
- [Asus WL-520gU - VPN + USB Printer server](#) USB section on the dd-wrt.wiki page about WL-520gU router
- [Example for WRT350n](#) Uses non-mega (to have plenty of space on jffs), jffs includes filesystem tools
- [Slim Samba install](#) Edit /etc/samba/smb.conf to fit your partitions (whole thing might fit on jffs if using non-mega dd-wrt on 8Mb flash routers)
- davidlaporte.org Asus WL-520gU with Optware Samba