

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

Contents

- [1 ??](#)
- [2 ??JFFS??? /jffs???](#)
 - ◆ [2.1 ????](#)
 - ◆ [2.2 Flash Wear: A Warning](#)
 - ◆ [2.3 Directions for \(normal\) users: using Web-GUI Interface](#)
 - ◆ [2.4 Directions for \(freaky\) Linux Gurus: CLI](#)
- [3 ??JFFS?????](#)
 - ◆ [3.1 Add a 1GB flash memory card](#)
 - ◆ [3.2 Add USB Storage](#)
 - ◆ [3.3 Using CIFS \(Client Samba\): Storing /jffs content remotely](#)
 - ◇ [3.3.1 Use a Startup Script: Example 1](#)
 - ◇ [3.3.2 Use Startup Scripts: Example 2](#)
 - ◇ [3.3.3 Use Startup Script: Example 3](#)
- [4 Testing ipkg](#)

??

Journalling Flash File System (JFFS/JFFS2) ?DD-WRT-enabled ??????????????.

- Located in */jffs*.

?????

- On DD-WRT Device main flash directly on the device (*/jffs*), or
- `mount --bind /storagelocation /jffs`.
Where *storagelocation* is
 - ◆ Off-board supplemental flash, see [SD/MMC mod](#), (*/mmc/jffs*).
 - ◆ Network Area Storage [NAS], or Server. (*//servername/sharename*)
 - ◆ JFFS reports as it's own free space, the free space on Share or SD Card, in the Web Interface

Examples of the kind of programs and data are:

- Custom configuration files
- NoCat splash pages and configuration
- [ipkg](#) (Requires JFFS. See [ipkg](#).)
- Web-site hosting
- Many other things.
- JFFS must be Enabled for certain /jffs-dependent applications.

??JFFS??? /jffs???

????

To enable JFFS you will need a router with at least 4MB flash, a build that supports JFFS such as the Mini build which is the smallest build to support JFFS thus leaving the most space for the JFFS partition, and your build must leave at least 324KB of flash free for the JFFS file system structure which will not be usable space. v24 pre-SP2 Standard builds no longer leave enough space for JFFS on 4MB devices so use the Mini build or one of Eko's specialty builds.

Very old versions of some Linksys routers (no longer manufactured for many years) contained 8MB of Flash and 32MB of RAM, many DD-WRT-capable devices today contain 4MB of Flash memory; a few models from certain manufacturers contain more flash (La Fonera, ASUS WL-500GP), some contain less (2MB). See Supported Devices.

Note that **DD-WRT v24sp1 VPN builds have the JFFS feature removed**, presumably due to conflicts that Brainslayer found between JFFS and bandwidth monitoring. See [this thread](#) for more information. Users interested in v24sp2 can opt:

1. Use a build other than VPN, both MINI and MEGA still have JFFS
2. Use an older build of DD-WRT
3. Use a custom build, either homemade or [dd-wrt.v24-10086_NEWD_openvpn_jffs_small.bin](#), which is described [here](#) as missing "snmp, all languages, some GUI styles and more."

NOTE FOR ASUS WL500W

As of October 7th, 2009, consider the following or you may BRICK you router:

Brainslayers mega flash size - 7.33MB

Eko's Big Flash size - 5.9MB

Now, the error only occurs when JFFS2 is enabled and the router reboots, so you are safe if you DONT enable JFFS2.

Total available JFFS2 size for BS's build is 640KB which overwhelms the kernel and causes a panic. Even using a storage device with the command mount -o bind /mnt/jffs /jffs will still cause a kernel panic.

Eko's JFFS2 partition size is 1640KB+, allowing much more room.

I tested this on BS's 12996,12874 and 12533 builds and partially bricked my router several times. (A simple power disconnect for 30 secs will allow it to recover)

Eko's build however works flawlessly AND provides the SAME features (except for connection warning) as BS's.

One thing I am still unsure of is why this particular model Asus router is subject to this bug, but in the mean time, I suggest using Eko's Big build until the dev's can work it out..

Flash Wear: A Warning

Take care not to store frequently changing files (i.e. often-changing log files) on Flash as this can cause flash wear-out over time.

- A journalling flash file system is specifically designed to minimize frequent overwrites.
- Files which are read many times but written infrequently work well on main flash.
- Using the space provided by SD/MMC_mod places wear on replaceable secondary flash storage.

Directions for (normal) users: using Web-GUI Interface

The steps to enable JFFS through the router web page are very specific. To avoid having to reset and reprogram your router, it's smart to make a backup here of your settings. If you follow these steps exactly, it should not lock up.

1. On the router web page click on *Administration*.
2. Scroll down until you see *JFFS2 Support* section.
3. Click *Enable JFFS*.
4. Click *Save*.
5. Wait couple seconds, then click *Apply*.
6. Wait again. Go back to the *Enable JFFS* section, and enable *Clean JFFS*.
7. **Do not click "Save"**. Click **Apply** instead.

The router formats the available space.

8. Wait till you get the web-GUI back, then disable "Clean JFFS" again.
9. Click "Save".
10. It may be wise to Reboot the router, just to make sure

Directions for (freaky) Linux Gurus: CLI

Enter the following commands from the CLI:

```
nvramp set jffs_mounted=1
nvramp set enable_jffs2=1
nvramp set sys_enable_jffs2=1
nvramp set clean_jffs2=1
nvramp set sys_clean_jffs2=1
nvramp commit
reboot
```

Note: Device might reboot/hang after cleaning which might take long; be patient.

WARNING

A user comments: I tried the previous commands under La Fonera firmware v24 RC4 and corrupted all my settings. I had to recover my unit with the procedures on La Fonera flashing page. I am not sure if this is a bug or these commands are not suited for La Fonera. Please try to use the web interface above. It works for me.

??? JFFS?

```
nvrans set sys_enable_jffs2=0
nvrans set sys_clean_jffs2=0
nvrans set jffs_mounted=0
nvrans commit
reboot
```

Afterwards you can delete these variables with "nvrans unset <variable>".

??JFFS?????

Add a 1GB flash memory card

- Add the [SD/MMC mod](#)
 - ◆ up to (at least) 1GB storage using common swappable SD memory cards (built-in support with DD-WRT v2.4 and later).
 - ◆ create the folder /mmc/jffs

```
mkdir /mmc/jffs
```
 - ◇ Use the Mount Bind command to point /jffs to the storage card:

```
mount --bind /mmc/jffs /jffs
```
 - ◇ with v24 enable JFFS2 support (tab: Administration -> tab: Management -> section: JFFS2 Support)
or command `nvrans set sys_enable_jffs2=1`
 - ◇ To use ipkg also create folder /jffs/tmp/ipkg

```
mkdir /jffs/tmp
mkdir /jffs/tmp/ipkg
```
- For Windows: [WinSCP](#) can be used to transfer files to and from the router, eliminating the need to pull the card out and put it in a card reader on a PC.
- For Linux, Mac OS X: Other SCP transfer applications exist.
- Removed cards can be accessed natively in Linux or OS X, or on Windows with the Ext2FS driver ([\[1\]](#) or [\[2\]](#)).
- On older firmware (v2.3), the limited JFFS space can be used to hold the drivers to load support for the SD card.

Add USB Storage

- Routers with USB storage like [Linksys WRTSL54GS](#) or [ASUS WL-500g Premium](#) can have firmware with built-in USB support and then /jffs mounted on the USB device. To enable USB on

??JFFS?????

these devices see: [USB storage](#)

- ◆ V24 and later firmware: USB support is available within Mega and some Mini builds.

- Using V24SP1 Mega on an Asus WL-500W with external USB HDD:

To mount the USB drive, and enable */jffs*

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

- Also enable JFFS2 support in either web-GUI (tab: Administration -> tab: Management -> section: JFFS2 Support)

or with the command

```
nvramp set sys_enable_jffs2=1
```

- To use ipkg also create folder */jffs/tmp/ipkg*

```
mkdir /jffs/tmp
mkdir /jffs/tmp/ipkg
```

Using [CIFS](#) (Client Samba): Storing */jffs* content remotely

One option is to create a share on the network and mount it on the router at */jffs* using [CIFS](#). Exact steps at [CIFS](#)

Overview:

- Enable JFFS in the web-GUI or put the following command in a Script or Startup Command.

```
nvramp set sys_enable_jffs2=1
```

- Create a share on a CIFS-capable server called *jffs*.
 - ◆ Standard Microsoft Windows Shared Folder
 - ◆ Other operating systems may refer to SMB or Samba
- In the Web GUI mount the [CIFS](#) share at */jffs*.
- Folder */jffs* appears locally, while

Content actually saved remotely, on

- ◇ NAS (Network Attached Storage)
- ◇ Server (PC or some type of file server)

- To use ipkg also create folder */jffs/tmp/ipkg*

```
mkdir /jffs/tmp
mkdir /jffs/tmp/ipkg
```

Costs:

- ! Setting up a NAS or Server only for router maybe not economical.
 - ◆ Equipment Costs: Requires existing NAS or PC Server
 - ◆ Electricity Costs: if entire Server is only serving router
- Availability Demands: CIFS share must be available
 - ◆ 24/7, or whenever the DD-WRT Device is on, and
 - ◆ Storing JFFS Share on quality equipment is required (i.e. Raid, Power Backup)
 - ◇ ! Remote storage down means /jffs down, could mean DD-WRT Device down (increases the weak points).
- Security Limitations: CIFS is limited to Microsoft's SMB protocol, offering up to only 128-bit security.
 - ◆ Poor Microsoft Protocol Security is not a problem if the LAN is physically and electronically protected from outside or inside intrusion (i.e. Guards, Guns, Locks, Dogs, Mines, Cameras, Firewalls)
- LAN Connection Required: NAS cannot be wirelessly attached to DD-WRT Device.
- Hassle: Using Microsoft Windows as the *jffs* Server
 - ◆ ! Symbolic links in */jffs* may not work.
 - ◆ ! Rename (mv) commands on the */jffs* mount may not work.

Most people only own an (inexpensive) Windows PC. A common Desktop PC does not a 24/7 Server make.

The following non-updated (as of 20080520) examples may be out of date:

- CIFS was broken on many builds this update author tried. Although other people have reported it working - It may be a limitation of *BusyBox* CIFS when using NTLMv2 and 128-bit encryption.
- (Mostly) only structural and spelling mistakes were corrected below (20080522)

Use a Startup Script: Example 1

1. Read how to use [Startup Scripts](#). Then place the following into your startup script to mount the JFFS partition:

```
umount /jffs #unmounts existing /jffs if existing
mount.cifs //<server's IP>/jffs /jffs -o username=<the username>,password=<the password>
nvram set sys_enable_jffs2=1 # fake-enable JFFS. Enable but do not commit. Makes ipkg think jffs2
```

NOTE: Username is *Guest* when hosted on Windows XP & 'Simple File Sharing' is enabled - This is the default on Windows XP Home.

Use Startup Scripts: Example 2

This method uses 2 scripts: The first will be started from */tmp/smbshare* & mount the samba share under */jffs* and start the other script. The second will be started from */jffs* and unmount */tmp/sambashare*.

Instructions:

Journalling_Flash_File_System/zh_cn

1. Enable CIFS (see also Jffs sharing in a Linux server).
2. Create a script called "startup.smb.sh" on the share:

```
#!/bin/sh

#smb settings: same as the settings in the webgui
# change these to mount a different share than the current at /jffs
SMBSHARE=`nvram get samba_share`
SMBUSER=`nvram get samba_user`
SMBPASS=`nvram get samba_password`

#script to start in /jffs once we're done mounting
JFFS_SCRIPT=startup.jffs.sh

#where to log what is happening
export SMBLOG=/tmp/smblog

##### ok, here we go... ;) #####
echo "" > $SMBLOG
cd /

#make sure JFFS is unmounted.
echo "umounting /jffs if necessary..." >> $SMBLOG
umount /jffs >> $SMBLOG 2>&1

#mount our smbshare at /jffs
echo "mounting $SMBSHARE at /jffs..." >> $SMBLOG
mount.cifs $SMBSHARE /jffs -o username=$SMBUSER,password=$SMBPASS >> $SMBLOG 2>&1

#hand over startup to the JFFS part
/jffs/$JFFS_SCRIPT & >> $SMBLOG 2>&1
```

4. create a script called "startup.jffs.sh" on the share:

```
#!/bin/sh

#wait a sec, just to be sure the smb startup script has quit
sleep 1

#unmount the smbshare, its identical to /jffs now anyway :)
umount /tmp/smbshare

#to enable ipkg, we fake JFFS enabled.
#since theres no commit, this wont actually create a /jffs at Startup.
nvram set sys_enable_jffs2=1

#ready to do specific stuff, like starting services
#or other commands from /jffs
```

5. in the webinterface, enable CIFS with appropriate share/user/pass settings from step 2.

- Enter "startup.smb.sh" as the script

6. Apply Changes
7. reboot

Now you should have the share configured in the web interface mounted at /jffs and should also be able to run 'ipkg' !

- NOTE: If you are a Linux newbie, and you have problems with file permissions, look at [this](#)
- Script files must be saved in UNIX format, not DOS. Windows editors that can do this are UltraEdit, EditPadPro and Crimson Editor, are free.
- [WinSCP](#) can be used to easily change the execute flags on the scripts.

Use Startup Script: Example 3

This method requires 2 shares available for DD-WRT, but allows you to have 2 different areas of storage for the router. Some might find it useful. It is an amalgamation of the previous 2 methods.

Things you need:

- 2 writeable shares on an NAS or Server.

Procedure:

1. Enable [CIFS](#)
2. Create a shell script in the network share, as follows:

```
#!/bin/sh
nvram set sys_enable_jffs2=1 # fake-enable JFFS
umount /jffs # unmount /jffs if it exists
mount.cifs //<server's IP>/<jffs share name> /jffs -o username=<your username>,password=<your pas
```

3. Enter the name of the script in the CIFS-Samba startup script box in the Web Interface.
4. Apply changes.
5. Reboot

As an example network setup:

- **/tmp/smbshare** located in a shared folder on the NAS called **ddwrt**, which contains scripts (including **startup.sh** containing the above commands), yet to be installed packages and other static data
- **/jffs** located in a shared folder called **jffs**, which contains program data for installed packages

Testing [ipkg](#)

Log in to the CLI (via Telnet or SSH) and enter the following commands:

```
cd /jffs
mkdir -p /jffs/tmp/ipkg
ipkg update
ipkg list
```

It is recommended that you reboot the router after running the above commands.