

Branslayer is working hard to merge the code base of DD-WRT with the OpenWRT firmware. This means that, for the most part, OpenWRT's package management system, ipkg, is available for DD-WRT. This allows for an easy way to add features not already in the DD-WRT firmware. Since ipkg is a command line program, you will need to either Telnet or SSH into your router to run this utility. You also need Jffs enabled.

Contents

- 1 Commandline Structure
 - ◆ 1.1 Usage
 - ◆ 1.2 Options
 - ◆ 1.3 Sub Commands
- 2 Finding Packages
- 3 Installing ipkg Packages
- 4 Required uClibc installation
 - ◆ 4.1 Setting LD_LIBRARY_PATH
- 5 Starting Programs/Daemons
- 6 What if your jffs partition is full (routers with NO JFFS space avail)
- 7 Use ipkg on DD-WRT 24
- 8 Ipkg on Startup
- 9 Mipsel versus other hardware
- 10 External Links

Commandline Structure

Usage

```
usage: ipkg [options...] sub-command [arguments...]
```

Options

```
-d <dest_name> or          Install, upgrade, or remove package from <dest_name>
-dest <dest_name>         where <dest_name> is ''either'' a folder path ''or'' a pre-defined
                           path in /etc/ipkg.conf
                           by default, ipkg on DD-WRT supports these pre-defined names:
                           root /jffs
                           ram /tmp

-o <offline_root>         Use <offline_root> as the root for offline installation.
-offline <offline_root>  where <offline_root> is a path

-force-dependends        Make dependency checks warnings instead of errors

-force-defaults          Use default options for questions asked by ipkg.
                           (no prompts). Note that this will not prevent
                           package installation scripts from prompting.
```

Sub Commands

update	Update list of available packages from <src> defined in /etc/ipkg.conf
upgrade	Upgrade all installed packages to latest version
install <pkg>	Download and install <pkg> (and dependencies)
remove <pkg>	Remove package <pkg>
list	List available packages and descriptions
files <pkg>	List all files belonging to <pkg>
search <file>	Search for a packaging providing <file>
info [pkg [<field>]]	Display all/some info fields for <pkg> or all
status [pkg [<field>]]	Display all/some status fields for <pkg> or all
depends <pkg>	Print uninstalled package dependencies for <pkg>

- <pkg> may be a package name, or a URI to the *.ipk or *.deb file
- arguments displayed in [] are optional

Finding Packages

OpenWRT.org hosts a list of [Official Packages](#), as well as a [Package Tracker](#) and openwrt.alphacore.net listing all known packages. Packages from the tracker will usually have to be installed by using <pkg> as a URI rather than a package name, unless you've updated your /etc/ipkg.conf file to point to a different [repository](#).

Another source of software for the dd-wrt is here ipkg.nslu2-linux.org, most of the files in this directory works fine on DD-WRT.

Installing ipkg Packages

The general steps for installing ipkg modules are as follows

-Pick a location to install to. Currently available locations are:

ram (/tmp)

root (the flash partition at [/jffs](#))

mmc (/mmc)

smbfs (/tmp/smbshare)

Then from the [command line](#) in the jffs directory, run the following commands:

```
ipkg update # pulls latest list of package listing from the default sites.
ipkg list # gives the list of ipkg's available
ipkg -d <location> install <packagename> # installs the package of your choice
```

If the package you want is not in the list, substitute the URI of the package for the <packagename> you want to install.

Packages installed to **ram** will be deleted on reboot. There is more ram than flash space, however.

[jffs](#) must be configured and initialized before packages can be installed to **root**

Ex: to install the noip package to *ram*:

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

ipkg_(tutorial)

```
ipkg update
ipkg list
```

Since noip is not in the list, see [Finding Packages](#)

```
ipkg -d ram install http://www.ramereh.net/openwrt/ipkg/noip\_1.6.0\_mipsel.ipk
```

Required uClibc installation

There is much confusion about LD_LIBRARY_PATH setting and some programs failing due to uClibc binary incompatibility between system provided uClibc libraries that DD-WRT provides in /lib/ directory and OpenWRT programs that require its own uClibc and is also provided as IPK package. For example <http://downloads.openwrt.org/whiterussian/packages/> To assure that program will run stable one must provide compatible uClibc library to the program. It is recommended to install OpenWRT uClibc manually with

```
cd /tmp
wget http://downloads.openwrt.org/whiterussian/packages/uclibc\_0.9.27-9\_mipsel.ipk
wget http://downloads.openwrt.org/whiterussian/packages/libgcc\_3.4.4-9\_mipsel.ipk
/bin/ipkg -force-depends install uclibc_0.9.27-9_mipsel.ipk libgcc_3.4.4-9_mipsel.ipk
```

This procedure overcomes installation of OpenWRT base-files package that are not required for running ipkg provided programs.

Setting LD_LIBRARY_PATH

Running OpenWRT programs also requires correct setting of the shared library search path. Default shared library search path is specified in /etc/ld.so.conf. But /etc/ld.so.conf is not used if LD_LIBRARY_PATH is set with /etc/profile. To see if LD_LIBRARY_PATH is set or no just enter

```
set
```

in console.

If unsure or if special paths are used one can also override /etc/ld.so.conf search paths with setting correct LD_LIBRARY_PATH. For example programs that reside in /jffs/bin or /jffs/sbin could be run with the following script openwrt-run.sh that should be located somewhere in the default search path.

```
#!/bin/sh
export PATH=/jffs/bin:/jffs/sbin:${PATH}
export LD_LIBRARY_PATH=/jffs/lib:/jffs/usr/lib:/jffs/usr/local/lib
$1 $2 $3 $4 $5 $6
```

One should use this script like

```
/jffs/bin/openwrt-run.sh swapon /dev/discs/disc0/part2
```

This will assure that program swapon will use correct uClibc library and not use paths provided in /etc/ld.so.conf

If you get an error like this:

Required uClibc installation

lpkg_(tutorial)

```
-sh: openwrt-run.sh: Permission denied
```

It means you haven't set file permission to allow execution. Navigate to your file in telnet/SSH, and do this:

```
chmod +x openwrt-run.sh
```

Now you should be able to run the script fine.

Starting Programs/Daemons

If you installed some daemons you need to manually activate them, because the openwrt /etc/init.d/ startup method is not (yet?) implemented. One possibility would be to

```
nvrnm set rc_startup=/jffs/etc/init.d/*
```

Or use the [Samba-Startup script](#).

For further Information see [Startup Scripts](#).

What if your jffs partition is full (routers with NO JFFS space avail)

You have to enable [jffs](#) anyway even if it is useless... and to create some required directory in your [Sambashare](#):

```
~# mkdir -p /tmp/smbshare/tmp/ipkg
```

Download then the packages directly to your [Samba](#) share and type:

```
~# ipkg -d smbfs install /tmp/smbshare/directory/package.ipk
```

or download and install at the same time with the same command:

```
~# ipkg -d smbfs install http://www.ramareth.net/openwrt/ipkg/noip\_1.6.0\_mipsel.ipk
```

You should see some warning messages like these ones:

```
ERROR: File not found: //jffs/usr/lib/ipkg/lists/whiterussian
You probably want to run `ipkg update'
ERROR: File not found: //jffs/usr/lib/ipkg/lists/non-free
You probably want to run `ipkg update'
```

which you can safely ignore. Also may be need command "mkdir -p /jffs/tmp/ipkg"

Use ipkg on DD-WRT 24

In my dd-wrt 24, I found cifs included, not smbfs or smbmount as told elsewhere on this page.

Here is how I partly solved it: On the Administration webpage, fill in the CIFS Automount options. Create a script with this and set it to be the startup script:

```
mount --bind /tmp/smbshare /jffs
nvram set sys_enable_jffs2=1
```

ipkg update failed for me, so I did a work-around:

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

I found *it's key when using ipkg and jffs* that you are in the jffs directory. In other words:

```
~# cd /jffs
```

```
/jffs # ipkg -d root install <pkg>.
```

(end of dd-wrt 24)

Ipkg on Startup

Since the available RAM is larger, is it possible to set up a startup script to install an ipkg? That way, each time it's rebooted the router will automatically download and install the ipkg we need.

One way to accomplish this is to write a startup script

Mipsel versus other hardware

DD-WRT now runs on a variety of hardware architectures. Some of the hardware that DD-WRT runs on is incapable of running software compiled for mipsel processors. Examples are Atheros WiSOC based hardware (e.g. LaFonera) and Intel Xscale IXP hardware (e.g. Actiontec MI424WR). If you see error messages that say "not found" or "syntax error" when you attempt to use the package you just installed, this may be your problem.

Unfortunately, as of build 13064, the ipkg.conf file in DD-WRT points to repositories that contain mipsel packages even if DD-WRT is running on incompatible hardware. If your hardware requires non-mipsel format packages you will not be able to use ipkg commands that search the web for packages (e.g. update, upgrade and search) unless you modify your ipkg.conf file. You must download the package you want and install it manually.

There are optware packages for a variety of hardware at <http://downloads.x-wrt.org/xwrt/kamikaze/snapshots/>.

ipkg_(tutorial)

Xscale packages: <http://downloads.x-wrt.org/xwrt/kamikaze/snapshots/ixp4xx/packages/>.

Atheros packages: <http://downloads.x-wrt.org/xwrt/kamikaze/snapshots/atheros/packages/>.

x86 packages: <http://downloads.x-wrt.org/xwrt/kamikaze/snapshots/x86/packages/>.

To install a package on non-mipsel hardware:

1) Download the .ipk file for the package you want to a convenient place on your router (e.g. /tmp). The file name will probably look like <packagename>_<version>_<hardwarename>.ipk.

2) Run ipkg -d <dest> install <filename>.ipk

External Links

[Guide for installing maiu IRC bouncer on the WRT using ipkg](#)