

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

You are here: [DD-WRT wiki mainpage](#) / [Scripting](#) / [Startup Scripts](#)

Contents

- [1 Introduction](#)
- [2 Web Interface Method of Loading Scripts onto Router](#)
 - ◆ [2.1 Prerequisites](#)
 - ◆ [2.2 Instructions](#)
 - ◆ [2.3 How it Works](#)
- [3 NVRAM Method](#)
 - ◆ [3.1 Prerequisites](#)
 - ◆ [3.2 Instructions](#)
 - ◆ [3.3 How it Works](#)
 - ◇ [3.3.1 Tip: Automating the Web-Interface](#)
- [4 Shell Script Method](#)
 - ◆ [4.1 Prerequisites](#)
 - ◆ [4.2 Instructions](#)
 - ◇ [4.2.1 Write the Shell Script](#)
 - ◇ [4.2.2 Save the Script](#)
 - ◇ [4.2.3 Make the Script Executable](#)
 - ◇ [4.2.4 Tell your scripts to load on startup, ipup or wanup](#)
- [5 Samba client method](#)
- [6 Links](#)

Introduction

Sometimes you install things and want them to load on startup. Writing startup scripts is the way to go!

Here are three ways to get dd-wrt to run them automatically.

Another method to have scripts run at startup is to place them in the right folder, with the .startup extension. Read more at [Script Execution](#)

Web Interface Method of Loading Scripts onto Router

Prerequisites

- Know how to use and access [Web Interface](#)
- Know the sequence of commands you'd like to run every startup

Startup_Scripts

- (optional) test commands at the command line to ensure correct operation

Instructions

1. Using the Web Interface, goto the **Administration** tab.
2. If available, go to the **Command** subtab, otherwise go to the diagnostics subtab.
3. Type the commands you wish to run every startup into the **Commands:** dialog box (place each command on a newline using the enter key, and if the command isn't something that stops running after a moment, put a '&' after the command)
4. Click the **Save Startup** button at the bottom of the page if you'd like the commands to save into the *rc_startup* variable (you may need to scroll). **Save Firewall** will save the commands into the *rc_firewall* variable.

How it Works

What this does is save your string of commands to the *rc_startup* variable in flash memory (also known as an NVRAM variable). Since flash memory doesn't get erased on startup, your "startup script" will remain with every boot and will always execute.

NVRAM Method

Prerequisites

- Know how to access and use Telnet or SSH
- Know the command(s) you would like to run every startup
- (optional) Test commands for correct operation.

Instructions

1. Login to your router using Telnet/SSH
2. To set the startup scripts, use the following:

```
nvramp set rc_startup="  
<command 1>  
<command 2>  
...  
<command n>  
"
```

When you give the command *nvramp commit* the *rc_startup* variable is saved in flash memory and will remain when you reboot.

```
nvramp commit
```

Startup_Scripts

A good trick for entering long scripts that, say you downloaded from the internet, is the Paste function of Windows Telnet.

At the telnet prompt enter:

```
nvramp set rc_startup=""
```

When it switches to the > prompt, copy the text from notepad or your web browser, or wherever your script is saved, then right-click the telnet logo at the top of the telnet window, and point to Edit, then click Paste. Voila! the entire script is typed out, with each command neatly placed on its own line. Now type the final " and hit enter to save.

How it Works

When you issue the first command you'll notice the prompt change to a ">" From here on every line you type is added to the *rc_startup* variable. **You are still technically typing on the same command line, even though you are using multiple lines** When you type the final "" and hit [Enter] you'll see the prompt change back and your first command has been executed. That is, your script has been added to the variable *rc_startup*.

At this point you can check the contents of *rc_startup* by issuing the command:

```
~ # nvramp get rc_startup
```

Tip: Automating the Web-Interface

For automating or testing the web interface, the free iMacros Firefox Add-On can be used. You can record and replay your activities inside the web interface. You can download it at <https://addons.mozilla.org/firefox/3863/>

Shell Script Method

A way to do this is to create a shell script that calls other shell scripts as required.

Simple nvramp scripts are fine for short scripts, but if you'd like to do a lot, or if you'd like to easily manage your startup programs, the best method is through using shell scripts.

Shell scripts make it easier to write long scripts that in turn *wget* executable files or scripts from the internet, or *ipkg -d ram install*, copy pre-made configuration files from the flash partition, etc., before finally running your program, and doing so on every startup.

This means you don't need to save as much stuff in router flash space (nvramp, which is rather limited) but you can instead fill up the much more prevalent RAM without having to manually re-install and configure your program(s) on every reboot.

And, since every program gets its own script, removing a program from the next boot is as easy as deleting, moving, or renaming its startup script file.

Prerequisites

- Know how to access and use Telnet or SSH
- Know the command(s) you would like to run at startup
- Know how to copy files to and from router using SCP, WinSCP or another method.
- Have jffs enabled and working properly.
- (Windows only) TextPad, Win32Pad, Notepad++ or another *nix friendly text editor. **Not notepad**
- (optional) Test your commands to ensure proper operation.

Instructions

Write the Shell Script

Write a shell script for each program you would like to run on startup.
your script should be a *.startup file

Shell scripts should look like this:

```
#!/bin/sh
<command 1>
<command 2>
<command 3>
...
killall <command n program name>
<command n>
```

Because of the design of the DD-WRT system, your shell script could end up being run periodically, and not just at boot (for example, all .wanup files get run multiple times at boot). This means that you'll need to ensure that only 1 copy of your program <command n> is ever run at a time. The *killall* command ensures that <command n> isn't running more than once simultaneously. This assumes that all commands that appear in the script prior to <command n> exit immediately after executing (such as *cp*, wget, or iptables.)

Ex a script: kismet_server.startup

```
#!/bin/sh
# Make the /tmp/usr/bin folder and move there
mkdir /tmp/usr
mkdir /tmp/usr/bin
cd /tmp/usr/bin

#copy the executable file from my home web server
wget http://192.168.1.2/kismet_server

#kill any previously-running instances of kismet_server
killall -q kismet_server

#execute kismet_server using /jffs/etc/kismet.conf as the configuration file.
/tmp/usr/bin/kismet_server -n -f /jffs/etc/kismet.conf
```

Save the Script

Enable JFFS to get some semi-permanent storage space. Save the script on the router in the */jffs/etc/config* directory. You may need to create this directory. Do so either in WinSCP or in a shell (SSH or Telnet) using the `mkdir` command.

Make the Script Executable

Making your script executable means changing the file rights to "700" Do this through WinSCP or with the command

```
~ # chmod 700 /jffs/etc/config/<scriptname>.startup
```

Tell your scripts to load on startup, ipup or wanup

For more info on this see article Script Execution

Samba client method

The drawback of jffs is the fact it doesn't work on some firmware versions and it is rather small. If you have a Samba or Windows server running on your LAN you could use the fabulous 'smbshare' option. This adds a vast amount of writable storage to your router, and it is possible to set an automatic startup script when the samba share mounts on boot. You can find the samba share options in the web interface under *administration*. If you set the startup script (as in Script Execution) option to *startup.sh* the actual script run after mounting will be */tmp/smbshare/startup.sh*

See also the Samba Filesystem for details.

Links

[LED Scripts](#)

[Script Execution](#)

[Useful Scripts](#)