

BootupAndShutdown < Webmin < TWiki

Bootup and Shutdown

This page explains methods via which servers and services are started at boot time, and tells you how to use Webmin to have your own commands run at startup.

Introduction to the Linux boot process

The very first thing to happen when a PC starts up is the loading of the BIOS from ROM. The BIOS (Basic Input/Output System) performs memory and other hardware checks, then loads a tiny piece of code from the first part of one of the system's hard disks, known as the master boot record or MBR. This piece of code is called a `_boot loader_`, and is responsible for displaying a menu of operating systems to the user and loading one of them. There are several boot loaders available for Linux such as LILO and GRUB, but they all do basically the same thing.

Once the kernel has been loaded, it mounts the root filesystem runs the `init` program, which is responsible for managing the rest of the boot process. It reads the `/etc/inittab` file and executes the commands it specifies, the most important of which begins execution of bootup scripts. Each of these scripts is responsible for a single task, such as initializing network interfaces, starting a webserver or mounting other filesystems. The scripts have a fixed order that they must execute in, as some of the later scripts are dependant on earlier ones - for example, network filesystems cannot be mounted until network interfaces have been enabled.

At shutdown time, a series of scripts is also run to shut down servers and un-mount filesystems. These scripts also have a fixed order, so that the de-activation of networking and other basic services happens last. If requested and supported by the hardware, the last step in the shutdown process will be the powering off of the system by the kernel.

When a Linux system starts up, different scripts are executed depending on which *runlevel* it is starting in. The runlevel can be set by the boot loader, or by the `/etc/inittab` file. The commonly used runlevels are :

5 - Graphical mode *All servers and services will be started, and X started to display a graphical login prompt on the console.

3 - Multi-user mode *All servers and services are started, but only the normal text login is available on the console.

2 - Multi-user mode without NFS *Almost all servers and services are started, but NFS filesystems are not mounted.

1 - Single user mode *Only the most basic system initialization is done, and a root shell opened on the console. This runlevel is useful if some bootup script is failing and making your system un-bootable.

See the section below on the **SysV Init Configuration module** for information on how to change the bootup runlevel.

The directory `/etc/rc.d/init.d` is usually used to store the actual bootup shell scripts. The scripts that are started or stopped in each runlevel are determined by symbolic links from the `/etc/rc.d/rc_X_d` directory, where `X` is the runlevel number. Each symbolic link has a name like `S_YY_scriptname`, in which `YY` is the order that the script is started in the boot process - the lower the number, the earlier the script starts. So `/etc/rc.d/rc5.d/S10network` would be run in runlevel 5 before `/etc/rc.d/rc5.d/S80sendmail`.

Not all Linux distributions use these directories for their bootup scripts. Some use `/etc/init.d` for the actual script files, while others (such as older versions of SuSE?) put everything in the `/sbin` directory. Fortunately, `/etc/rc.d` seems to be becoming the standard base directory in newer distributions. Of course, if you are using Webmin you don't have to worry about the locations of any of these directories as it always knows where they are.

The Bootup and Shutdown module

This module allows you to create and edit the scripts that are run at bootup and shutdown time, called actions by the module. It can be found under the System category in Webmin, and when you enter it the main page will display a list of all available actions, whether they are started at boot, and a short description for each. See this image for an example.

The screenshot shows the Webmin interface for the 'Bootup and Shutdown' module. On the left is a navigation menu with categories like System, Servers, Networking, Hardware, Cluster, and Others. The main content area is titled 'Module Config Bootup and Shutdown' and contains a table of actions. The table has four columns: Action, Start at boot?, Order, and Description. The 'dansguardian' row is highlighted in yellow.

Action	Start at boot?	Order	Description
<input type="checkbox"/> acpid	No	08	Advanced Configuration and Powermanagement
<input type="checkbox"/> adsl	No	70	PPPoE for ADSL
<input type="checkbox"/> amd	Yes	30	Auto Mount Daemon (NFS & local)
<input type="checkbox"/> amd.net	No	99	Auto Mount Daemon (slave for /net NFS)
<input type="checkbox"/> atalk	No	35	AppleTalk Services (atalk)
<input type="checkbox"/> atd	Yes	41	Batch Server (atd)
<input type="checkbox"/> autodialer	No	50	Automatic Dialin Demon
<input type="checkbox"/> bigfs	No	99	Very large filesystem support (delayed & backgrounded)
<input type="checkbox"/> cameleo	No	90	Cameleo Servers
<input type="checkbox"/> cfengine	No		starts cfd
<input type="checkbox"/> cron	Yes	40	Cron daemon
<input type="checkbox"/> csmnd	No	50	CSM Client
<input type="checkbox"/> cups	Yes	35	Print server (CUPS)
<input checked="" type="checkbox"/> dansguardian	No		Startup script for dansguardian
<input type="checkbox"/> dhcpd	Yes	56	DHCP and BOOTP boot server
<input type="checkbox"/> dnscache	No	22	DNS Forwarder
<input type="checkbox"/> docview	No	86	Documentation web server
<input type="checkbox"/> dovecot	Yes	99	Start dovecot IMAP and POP3 server
<input type="checkbox"/> firewall	No	99	Load firewall rules
<input type="checkbox"/> frox	No		Start the transparent FTP proxy frox
<input type="checkbox"/> fudusync	Yes	99	Update DNS with fudu IP
<input type="checkbox"/> gpm	No	75	GPM Console Mouse Driver
<input type="checkbox"/> halt	No	99	halt/reboot/poweroff
<input type="checkbox"/> heartbeat	No		Start high-availability services
<input type="checkbox"/> hostsentry	No		Start the hostsentry LAD tool
<input type="checkbox"/> hotplugd	No	08	USB Hotplugging Daemon
<input type="checkbox"/> httpd	Yes	85	Web Server
<input type="checkbox"/> inet	Yes	16	Basic IP service dispatcher (inetd -- normally enabled!)
<input type="checkbox"/> ipsec	No		IPsec startup and shutdown script
<input type="checkbox"/> iptables	No		Startup script to implement /etc/sysconfig/iptables pre-defined rules.
<input type="checkbox"/> ipx	No	26	Internetwork Packet eXchange (ipx)
<input type="checkbox"/> irda	No	07	IrDa utilities
<input type="checkbox"/> jabber	No		This shell script takes care of starting and stopping jabber. (C) 2000 IDEALX.

The Bootup and Shutdown module

Each Linux distribution has its own set of standard action scripts, so on one system the script httpd may start the Apache webserver, but on another it may be called apache. You should be able to get a good idea of what each script does from its description though.

Configuring an action to start at bootup

If some server on your system such as Apache or Squid is not currently being started at boot time, you can use this module to change that. On most Linux distributions, every server that comes with the distribution will have its own bootup action script, but not all will be enabled by default. To configure an action to start at boot time, the steps to follow are :

1. On the main page of the module, click on the name of the action that you want to enable. This will take you to the action editing page, as shown in the image below.
2. Change the **Start at boot time?** option from **No** to **Yes**.
3. Click the **Save** button, and your browser will return to the list of actions on the main page.

The screenshot shows the 'Edit Action' page in Webmin. On the left is a navigation menu with categories like 'Login: screenshots', 'System', 'Servers', 'Networking', etc. The main content area is titled 'Edit Action' and shows details for the 'httpd' action. The 'Name' field contains 'httpd'. The 'Action Script' field contains a shell script with the following content:

```
#!/bin/bash
#
# $Id: apache.init,v 1.1 2000/11/09 11:57:20 sps Exp $
#
### BEGIN INIT INFO
# Provides:
# Required-Start: $network
# Required-Stop: $network
# Default-Start: 3 4 5
# Default-Stop: 0 1 2 6
# Description: httpd
#           Bring up/down the Apache web server daemon.
### END INIT INFO
#
# Written by Miguel van Spaendonck <miguels@drinkel.ov.org>.
# Modified for Debian GNU/Linux by Ian Murdock <imurdock@gnu.ai.mit.edu>.
```

Below the script, the 'Start at boot time?' option is set to 'Yes' (radio button selected). At the bottom of the form are buttons for 'Save', 'Start Now', 'Restart Now', 'Show Status', 'Stop Now', and 'Delete'. A link 'Return to bootup and shutdown actions' is also visible.

The action editing form

If there is a server that is currently being started at boot time that you want to disable, just follow the same steps but set the **Start at boot time?** option to **No** instead.

Starting and stopping actions

Even though action scripts are normally started at boot time and stopped at shutdown time, you can start or stop them at any time using Webmin. Many action scripts can also perform additional functions, such as showing the status of a server or reloading its configuration. To start or stop an action, do the following :

1. On the main page of the module, click on the name of the action. This will take you to the action

editing form shown in the image below.

2. At the bottom of the page in the middle will be a row of buttons, each for running the action script to perform some function. Depending on the script there may be different buttons available, but some of the most common are : *Start Now *Immediately starts the server or service. On some versions of Linux, this will do nothing if the action has already been started and the server is already running. *Stop Now *Stops the server or service. Of some Linux versions, this will do nothing unless the action has already been started. *Restart Now *Stops and restarts the server. In many cases, this will do nothing if the action has not been started yet. *Reload Now *Where available, this function tells the server started by the action to re-read its configuration files. *Show Status *Just displays a message telling you if the server is running or not, and if so what its PID is.
3. After you click the button for the function that you want to perform, a page showing the output from the action script will appear. This should indicate whether the action was performed successfully or not.

Adding a new action

If you have a command that you want run at boot time, creating a new action script is the best way to set it up. Servers like Apache or Qmail that have been compiled and installed manually do not have actions, so you will need to create one that runs whatever command is necessary to start the server.

To create your own action, follow these steps :

1. On the main page of the module, click the **Create a new bootup and shutdown action** link above or below the list of existing actions. This will take you to the form shown in Figure 9-3 for entering the code for your new action script.
2. In the **Name** field, enter a short name for the action like *qmail*. Every action must have a unique name.
3. In the **Description** field, enter a few lines of text to describe your action - maybe something like *Start the Qmail mail server*. This will show up on the main page of the module under the **Description** column.
4. The **Bootup commands** field must be filled in with the shell commands that you want run when your action is started at boot time. For example, if you wanted to start Qmail you might enter */var/qmail/rc*.
5. The **Shutdown commands** field should be filled in with commands that you want run when your action is stopped. For example, to stop Qmail you might enter *killall -9 qmail-send*.
6. Assuming you want your action run at boot time, set the **Start at boot time?** option to **Yes**.
7. Finally, click the **Create** button to save the new action. Webmin will create a script in the */etc/rc.d/init.d* directory combining the commands you entered with a standard wrapper to make a valid action script. Your action will be set to run in the current runlevel, with order number 99 so that it is run last. If you want to be able to control exactly which runlevels and in what order your action is run, see the **Allow selection of individual runlevels** option in the **Module configuration** section below.

The screenshot shows the 'Create Action' form in Webmin. The form is titled 'Create Action' and is part of the 'Module index'. It contains several text input fields for 'Name', 'Description', 'Bootup commands', and 'Shutdown commands'. There is a 'Start at boot time?' checkbox with 'Yes' selected. A 'Create' button is at the bottom left, and a 'Return to bootup and shutdown actions' link is at the bottom right. A sidebar on the left shows a navigation menu with categories like System, Servers, Networking, Hardware, Cluster, and Others.

The action creation form

After an action has been created, you can edit the start and stop commands by following this process :

1. On the main page of the module, click on the name of your action. This will take you to the action editing form shown in Figure
2. -2.
3. In the **Action script** text box, look for a line like 'start'). The commands that are run at boot time will after it, up the a line containing just ;;. Edit them as you wish, but leave the surrounding code alone. Similarly, the commands that are run when the action is stopped are between 'stop') and ;;. Changing any other part of the script is a bad idea unless you know what you are doing.
4. Click the **Save** button to apply your changes.

Any of the existing action scripts can be edited using Webmin, not just your own creations. Be careful editing them, as they may have a format totally different to the scripts created by Webmin.

Rebooting or shutting down your system

Linux systems should always be rebooted or shut down using the appropriate commands, rather than simply turning off the power or hitting the reset button. If not, you may lose data on your local hard drives and will certainly have to wait through a lengthy filesystem check with fsck at boot time if using a non-journaling filesystem.

To reboot, simply do the following :

1. At the bottom of the main page of the Bootup and Shutdown module, click the **Reboot System** button. This will take you to a page confirming if you really want to reboot.
2. Click the **Reboot System** button on the confirmation page. The shutdown process will start immediately, and if you are logged in at the console your session will be logged out. After all the

shutdown scripts have been run, the system will bootup again as explained in the introduction.

The process for shutting down is almost identical - just use the **Shutdown System** button at the bottom of the page instead.

Configuring the Bootup and Shutdown module

Like most modules, Bootup and Shutdown can be configured by clicking on the **Module Config** link on the main page. This will take you to the standard configuration editing page, on which the following settings are available under the **Configurable options** header :

None of the other options on the configuration page should be changed, as they are set automatically by Webmin based on your operating system type.

Other operating systems

The system of bootup scripts used by Linux is used by many other Unix operating systems, but not all of them. Even those that do use it have some slight differences in their implementation, and almost all use different directories for storing the actual scripts and links.

Sun Solaris, HP/UX, SCO UnixWare? , SCO OpenServer? , Compaq Tru64/OSF1 *and SGI Irix
*All these operating systems use action scripts that are very similar to Linux, but are stored in different directories. Because those that come with the system do not have descriptions the main page of the module will just display action names by default.

*FreeBSD, NetBSD and *OpenBSD *The BSD family of operating systems does not use action scripts at all, instead relying on a fixed set of scripts that are run at boot time. One of these scripts (/etc/rc.local) is for system administrators to add their own commands to be run at boot time. On any of these operating systems, the main page of the module will just display a form for editing the rc.local file, above the **Reboot System** and *Shutdown System* buttons. To add any commands that you want run at boot time, just enter them into the text box and click the **Save** button.*

**IBM AIX *AIX is very similar to the BSD operating systems in that it does not have action scripts. Instead, the file /etc/rc can be edited to add additional commands to be run at boot time, using the form on the main page of the module.*

*Apple MacOS? X *Apple's version of Unix uses a totally different set of files for storing actions to be run at boot time than any other supported operating system. Separate action scripts still exist, but the user interface in this module for viewing and editing them is quite different.

If your operating system is not on the list above then it is not supported by the Bootup and Shutdown module at all, so the module icon will not appear in Webmin.

The SysV? Init Configuration module

As explained in the introduction to this chapter, the very first file read by the system to determine which

commands to run at boot time is `/etc/inittab`. It is read by the `init` program, which is the first process to be run after the Linux kernel finishes loading, and which remains running until the system is shut down. The `inittab` file specifies which runlevel to boot into, the commands to be run to start all of the action scripts, processes to start to display text and graphical login prompts, and commands to run in the case of an impending power failure.

The SysV? Init Configuration module, found under the System category in Webmin, can be used to edit any of these commands. However, as they are critical for ensuring that your system boots up properly, editing them is a bad idea unless you really know what you are doing. The only thing that you might want to change is the bootup runlevel, so that your system does not display an unnecessary graphical login prompt if it is not needed.

To change the initial runlevel, the steps to follow are :

1. Enter the SysV? Init Configuration module. The main page will display a list of commands and the runlevels and situations in which they are executed, as shown in the image below.
2. Click on the entry in the **ID** column for the row in which the **Action** is **After system boot**. This will take you to a form for editing the `inittab` file entry.
3. For the **Bootup runlevel** option, de-select whichever level is currently selected and choose a new one. Make sure that you choose exactly one level, such as **3** (for text login mode) or **5** (for graphical login mode). See the explanation in the introduction to this chapter for details on what each runlevel means.
4. Click the **Save** button to have your change written to the `inittab` file. The browser will return to the main page of the module.
5. If you like, you can reboot the system now using the Bootup and Shutdown module.

Login: screenshots
 Webmin
 System
 Bootup and Shutdown
 Change Passwords
 Disk Quotas
 Disk and Network Filesystem
 Filesystem Backup
 LDAP Client
 LDAP Users and Groups
 Log File Rotation
 MIME Type Programs
 MON Service Monitor
 PAM Authentication
 Running Processes
 Scheduled Commands
 Scheduled Cron Jobs
 Security Sentries
 Software Packages
 SysV Init Configuration
 System Documentation
 System Logs
 System Logs NG
 Users and Groups
 Servers
 Networking
 Hardware
 Cluster
 Others
 System Information
 Logout

Help..
 Module Config

Search Docs..

SysV Init Configuration

Select all. | Invert selection. | Create a new init process.

ID	Active?	Bootup runlevel	Action	Process
id	Yes	3	After system boot	
s0	Yes	None	During system boot	/bin/bash -c 'C=/sbin/booterd: [-x \$C] && \$C'
si	Yes	None	During system boot	/bin/bash -c 'C=/etc/rc.d/rc.modules: [-x \$C] && \$C default'
s2	Yes	None	During system boot	/bin/bash -c 'C=/etc/rc.d/rc.serial: [-x \$C] && \$C'
bw	Yes	None	Boot time, wait until termination	/etc/rc.d/rc.boot
~1	Yes	5	Wait	/etc/rc.d/rc 1
~~	Yes	5	Wait	/sbin/sulogin
0	Yes	0	Wait	/etc/rc.d/rc 0
1	Yes	1	Wait	/etc/rc.d/rc 1
2	Yes	2	Wait	/etc/rc.d/rc 2
3	Yes	3	Wait	/etc/rc.d/rc 3
4	Yes	4	Wait	/etc/rc.d/rc 4
5	Yes	5	Wait	/etc/rc.d/rc 5
6	Yes	6	Wait	/etc/rc.d/rc 6
z6	Yes	6	Respawn process	/sbin/sulogin
ca	Yes	1, 2, 3, 4, 5	Ctrl-Alt-Del	/sbin/shutdown -t3 -r now
kb	Yes	None	kbrequest	/bin/echo "Keyboard Request--edit /etc/inittab to let this work."
pf	Yes	None	Power goes down without waiting	/sbin/shutdown -h +5 "Power Failure: System Shutting Down"
p1	No	None	Power fail	/sbin/shutdown -c 2> /dev/null
p2	No	None	Power fail	/sbin/shutdown -h now "Battery Low..."
po	Yes	1, 2, 3, 4, 5	Power is restored	/sbin/shutdown -c "Power Restored: Shutdown Cancelled"
gu	Yes	5	Respawn process	/bin/sh -c 'C=/etc/rc.d/rc.gui:[-x \$C]&&exec \$C:init 3'
0	Yes	2, 3, 4, 5	Once time at level	/usr/sbin/ttyload
1	Yes	1, 2, 3, 4, 5	Respawn process	/sbin/getty tty1 VC linux
2	Yes	2, 3, 4, 5	Respawn process	/sbin/getty tty2 VC linux
3	Yes	2, 3, 4, 5	Respawn process	/sbin/getty tty3 VC linux
4	Yes	2, 3, 4, 5	Respawn process	/sbin/getty tty4 VC linux
5	Yes	2, 3, 4, 5	Respawn process	/sbin/getty tty5 VC linux
6	Yes	2, 3, 4, 5	Respawn process	/sbin/getty tty6 VC linux
7	No	2, 3, 4, 5	Respawn process	/sbin/getty ttyS0 VC vt100
8	No	2, 3, 4, 5	Respawn process	/sbin/getty ttyS1 VC vt100
mo	No	2, 3, 5	Respawn process	/usr/sbin/mgetty -s 3B400 modem
mo	No	3, 5	Respawn process	/usr/lib/fax/faxgetty /dev/modem
l6	No	3, 5	Respawn process	/usr/sbin/mboxgetty -d /dev/ttyl6

Done

The SysV? Init Configuration module

The module is also available on the Solaris, HP/UX, UnixWare? , OpenServer? , AIX and Irix operating systems. Its basic structure and purpose is the same on all systems, but the actual default commands will differ significantly. However, the instructions above for changing the bootup runlevel will work on all operating systems.