Fedora 22

Installation Guide

Installing Fedora 22 on 32 and 64-bit AMD and Intel

Fedora Documentation Project

Fedora 22 Installation Guide
Installing Fedora 22 on 32 and 64-bit AMD and Intel
Edition 1

Author Fedora Documentation Project

This manual explains how to boot the Fedora installation program, **Anaconda**, and how to install Fedora 22 on 32 and 64-bit AMD and Intel systems. It also covers advanced installation methods such as automated Kickstart installations, booting the installation from a network location, remote access to the installation system using VNC, and system upgrades from previous versions of Fedora. It also describes common post-installation tasks and explains how to troubleshoot common issues related to the installation.

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts*[1] set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to the first virtual terminal. Press **`Ctrl`**+**`Alt`**+**`F1`** to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click

[1] https://fedorahosted.org/liberation-fonts/

> **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.
>
> The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.
>
> To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;
```

```
public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: *http://bugzilla.redhat.com/bugzilla/* against the product **Fedora.**

When submitting a bug report, be sure to mention the manual's identifier: *install-guide*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

3. Acknowledgments

Certain portions of this text first appeared in the *Red Hat Enterprise Linux Installation Guide*, copyright © 2014 Red Hat, Inc. and others, published by Red Hat at *https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/*.

Introduction

This guide covers installation of Fedora, a Linux distribution built on free and open source software. This manual helps you install Fedora on desktops, laptops, and servers. The installation system is easy to use even if you lack previous knowledge of Linux or computer networks. If you select default options, Fedora provides a complete desktop operating system, including productivity applications, Internet utilities, and desktop tools.

This document details the full range of installation options, including those that apply only in limited or unusual circumstances. Understanding of all topics described in this document is not necessary to successfully perform the installation in most cases.

1.1. Background

1.1.1. About Fedora

To find out more about Fedora, visit the *Fedora Project Website*[1]. Other documentation describing additional topics related to Fedora is available at *Fedora Documentation*[2]. Also see the *Fedora Project Wiki*[3].

1.1.2. Getting Additional Help

If you encounter any problems which are not described in documentation, you might get help from members of the community - developers, users, and others. There are many ways to get help: the Ask Fedora website, mailing lists, forums, or IRC. For a summary of available resources, see the *Communicating and Getting Help*[4] page on the Fedora wiki.

1.2. About This Document

1.2.1. Goals

This guide helps a reader:

- Understand how to locate the Fedora distribution online
- Create configuration data that allows a computer to boot Fedora
- Understand and interact with the Fedora installation program
- Complete basic post-installation configuration of a Fedora system

[1] http://fedoraproject.org/
[2] http://docs.fedoraproject.org/
[3] https://fedoraproject.org/wiki/Fedora_Project_Wiki
[4] https://fedoraproject.org/wiki/Communicating_and_getting_help

Note

This guide does not cover *use* of Fedora. To learn how to use an installed Fedora system, see the other manuals available at *Fedora Documentation*[5].

1.2.2. Target Audience

This guide is intended for Fedora users of all levels of experience. However, it describes the installation process and its many options in far greater detail than most users are likely to require. You do not need to read and understand this entire document to install Fedora on a computer. This document is most likely to help experienced users perform advanced and unusual installations.

[5] http://docs.fedoraproject.org/

Downloading Fedora

Fedora Flavors

Fedora provides three primary flavors tailored for some specific use cases. *https://getfedora.org* offers Fedora Cloud for scalable infrastructure, Fedora Server for organizational infrastructure, and Fedora Workstation for the developer and desktop user.

For alternative desktop environments or media built for more niche purposes, check out *Fedora Spins*[1].

Each of these downloads provides a different set of default packages, but you can add to your system after the initial installation to customize it for your needs. The installation process is the same for all spins and flavors, so you can use this guide for any choice you make.

Which Architecture Is My Computer?

Most modern systems are 64 bit x86 architecture. If your computer was manufactured after 2007, or you aren't sure, you probably have a **x86_64** system.

Changing a Fedora installation from one architecture to another is not supported. Use the following table to determine the architecture of your computer according to the type of processor. Consult your manufacturer's documentation for details on your processor, or resources such as *http://ark.intel.com/* or *http://products.amd.com/*, if necessary.

Table 2.1. Processor and architecture types

Processor manufacturer and model	Architecture type for Fedora
some Intel Atom, Core series, Pentium 4, and recent vintage Xeon; AMD Athlon, Duron, some Semprons; and older; VIA C3, C7	i386
some Intel Atom, Core 2 series, Core i series and Xeon; AMD: Athlon 64, Athlon II, Sempron64, Phenom series, Fusion series, Bulldozer series and Opteron; Apple MacBook, MacBook Pro, and MacBook Air	x86_64

Media Types

Several media types are available. Choose the one that best suits your requirements.

Live Image

Live images allow you to preview Fedora before installing it. Instead of booting directly into the installer, a live image loads the same environment you'll get after installation. Fedora Workstation and Fedora Spins are live images.

Use a live image to install your favorite system, test Fedora on new hardware, troubleshoot, or share with friends.

DVD Image

DVD images boot directly into the installation enviroment, and allow you to choose from a variety of packages that are provided with it. In Fedora 21, the DVD option is only available in the Fedora Server flavor.

[1] https://spins.fedoraproject.org

Use the Fedora Server DVD image when you want customized Fedora Server installations using an offline installation source.

netinstall Image

The netinstall image boots directly into the installation environment, and uses the online Fedora package repositories as the installation source. With a netinstall image, you can select a wide variety of packages to create a customized installation of Fedora.

The Fedora Server netinstall image is a universal one, and can be used to install any Fedora flavor or your own set of favorite packages.

ARM images

For many ARM systems, Fedora provides preconfigured filesystem images. Write the image to removable media and boot directly into a Fedora installation that's ready to use.

ARM devices often require special setup procedures that aren't covered in this guide. Start learning about Fedora ARM at *https://fedoraproject.org/wiki/Architectures/ARM*

Cloud Images

Fedora Cloud images are preconfigured filesystem images with very few packages installed by default. They include special tools for interacting with cloud platforms, and are not intended to be used outside of cloud environments.

Fedora Cloud comes in several varieties. The Fedora Cloud Base image is a minimal base for cloud deployments. The Fedora Cloud Atomic image is a Docker container host that uses *Project Atomic*[2] technology for updates. A Docker base image for Fedora is also available.

Cloud images are preconfigured and do not require installation as described in this guide. Get started using Fedora Cloud at *http://fedoraproject.org/wiki/Cloud*

Boot Images

The tiny images at *https://boot.fedoraproject.org/* are written to CDs, USB drives, or even floppy disks. The BFO image loads installation media from Fedora's servers and directly loads an installation environment, like the netinstall ISO.

BFO images work like PXE deployments, without having to set up a server.

[2] http://www.projectatomic.io/

Part I. Installing Fedora

This part of the *Fedora Installation Guide* details the installation process itself, starting with the steps you must take to prepare for the installation, and ending with the steps you need to take immediately after the installation finishes and your system reboots for the first time. It also includes a section about troubleshooting common problems which may appear before, during or immediately after the installation.

Preparing for Installation

This chapter describes the steps you need take before you begin the installation. Not every step must be strictly followed - for example, if you plan to use the default installation settings, you do not need to gather system information such as disk device labels/UUIDs or network information such as the system's IP address. However, you should still go through this chapter, as it also describes the available types of installation media and how to prepare boot media and installation sources.

3.1. Upgrade or Install?

If you already have Fedora installed and want to upgrade your installation to the current version, there are two basic ways to do so:

Automatic upgrade using **FedUp**

> The preferred way to upgrade your system is an automatic upgrade using the **FedUp** utility. For information on performing an automatic upgrade, see *Section 12.1, “Automatic System Upgrade Using FedUp”*.

Manual Reinstallation

> You can upgrade to the latest version of Fedora manually instead of relying on **FedUp**. This involves booting the installer as if you were performing a clean installation, letting it detect your existing Fedora system, and overwriting the root partition while preserving data on other partitions and volumes. The same process can also be used to reinstall the system, if you need to. For detailed information, see *Section 12.2, “Manual System Upgrade or Reinstallation”*.

Warning

Always back up your data before performing an upgrade or reinstalling your system, no matter which method you choose.

3.2. Downloading Boot and Installation Images

The Fedora Project offers different flavors for tailored for some specific use cases. Choose the Fedora flavor best for you, or you can build your own by customizing after the installation, or by using a kickstart file as described in *Section 9.1.1, “Creating a Kickstart File”*.

Read more about Fedora Workstation, Fedora Cloud, Fedora Server and the available media types in *Chapter 2, Downloading Fedora*.

You can also choose a Fedora *Spin* featuring favorite alternative desktops or tools for specialized tasks at *http://spins.fedoraproject.org*.

3.3. Verifying the Downloaded Image

Because transmission errors or other problems may corrupt the Fedora image you have downloaded, it is important to verify the file's integrity. After the images are created, an operation is performed on the file that produces a value called a **`checksum`** using a complex mathematical algorithm. The operation is sufficiently complex that *any change* to the original file will produce a different checksum.

By calculating the image's **`checksum`** on your own computer and comparing it to the original **`checksum`**, you can verify the image has not been tampered with or corrupted. The original checksum values are provided at *https://fedoraproject.org/verify*, and are **gpg** signed to demonstrate their integrity.

3.3.1. Verifying checksums on Windows systems

Procedure 3.1. Verifying checksums on Windows systems

1. Download the Fedora image of your choice from *https://fedoraproject.org/get-fedora* and the corresponding checksum file from *https://fedoraproject.org/verify*

2. Open a **`powershell`** session.

3. Change to the directory containing the downloaded files.

   ```
   > cd $HOME\Downloads\
   > ls

   Directory: C:\Users\Pete\Downloads

   Mode                LastWriteTime         Length  Name
   ----                -------------         ------  ----
   -a---        11/25/2014  12:39 PM            272  Fedora-Server-21-x86_64-CHECKSUM
   -a---        11/25/2014  12:39 PM     2047868928  Fedora-Server-DVD-x86_64-21.iso
   ```

4. Load the resources required to calculate the checksum.

   ```
   > $image = "Fedora-Server-DVD-x86_64-21.iso"
   > $checksum_file = "Fedora-Server-21-x86_64-CHECKSUM"
   > $sha256 = New-Object -TypeName
    System.Security.Cryptography.sha256CryptoServiceProvider
   > $expected_checksum = ((Get-Content $checksum_file | Select-String -Pattern $image) -
   split " ")[0].ToLower()
   ```

5. Calculate the downloaded image's checksum. This will take a while!

   ```
   > $download_checksum =
    [System.BitConverter]::ToString($sha256.ComputeHash([System.IO.File]::ReadAllBytes("$PWD
   \$image"))).ToLower() -replace '-', ''
   ```

6. Compare the calculated checksum to the expected checksum.

   ```
   > echo "Download Checksum: $download_checksum"
   > echo "Expected Checksum: $expected_checksum"
   > if ( $download_checksum -eq "$expected_checksum" ) {
   echo "Checksum test passed!"
   } else {
   echo "Checksum test failed."
   }
   ```

3.3.2. Verifying checksums on Linux and OSX systems

Procedure 3.2. Verifying checksums on Linux and OSX systems

1. Download the Fedora image of your choice from *https://fedoraproject.org/get-fedora* and the corresponding checksum file from *https://fedoraproject.org/verify*

2. Open a terminal window, and navigate to the directory with the downloaded files.

   ```
   $ cd ~/Downloads
   ```

3. Use the appropriate utility to verify the image checksum.

   - For Linux:

     ```
     $ sha256sum -c *CHECKSUM
     ```

   - For OSX:

     ```
     $ shasum -a 256 -c *CHECKSUM
     ```

3.4. Preparing Boot Media

Fedora images are **`Hybrid ISOs`** and can be used to create installation media with both optical and USB disks, for booting on both BIOS and UEFI systems.

Note on Universal USB Creators

Universal USB creation tools such as **Unetbootin** are a historically popular way to create USB installers from ISOs intended for optical media. They typically function by creating a filesystem on the USB drive, extracting files from the image, and writing **syslinux** bootloader to the device.

These methods circumvent the bootloader configuration built into Fedora images, which are pre-partitioned and designed to boot on UEFI systems with SecureBoot enabled as well as BIOS systems. They do not produce a consistent result with Fedora's images, especially for use with UEFI systems.

Utilities that use a direct write method, and do not modify the Fedora image, will produce the most consistently successful results.

Be sure you choose the right device!

Most media creation methods in this section are destructive. Ensure you do not need any data on the USB stick, and double check you have chosen the correct device before continuing.

Procedure 3.3. Creating USB Media on Windows

1. Download and run the ***SUSE Studio ImageWriter***[1] or ***Rawrite32***[2].

2. Choose the downloaded Fedora image. You may have to change the file selection dialog's options to view ISO or all file types.

3. Choose the USB device you wish to convert to installation media.

4. Continue with media creation using the appropriate button. After a few minutes, it will report the process is complete and your installation media will be ready to use.

Procedure 3.4. Creating USB media with **GNOME Disks**

1. On a system with **GNOME**, or with the *gnome-disk-utility* package installed, open **Disks** using the system menu.

2. Click your USB device in the left column.

3. Click the menu icon in the upper right corner of the window, and choose the **Restore Disk Image** option.

4. Navigate to your image file and click **Start Restoring**. After a few minutes, it will report the process is complete and your installation media will be ready to use.

Procedure 3.5. Creating USB Media on the Linux command line

1. Open a terminal window and insert the usb drive.

2. Find the `device node` assigned to the drive. In the example below, the drive is given **sdd**.

```
$ dmesg|tail
[288954.686557] usb 2-1.8: New USB device strings: Mfr=0, Product=1, SerialNumber=2
[288954.686559] usb 2-1.8: Product: USB Storage
[288954.686562] usb 2-1.8: SerialNumber: 000000009225
[288954.712590] usb-storage 2-1.8:1.0: USB Mass Storage device detected
[288954.712687] scsi host6: usb-storage 2-1.8:1.0
[288954.712809] usbcore: registered new interface driver usb-storage
[288954.716682] usbcore: registered new interface driver uas
[288955.717140] scsi 6:0:0:0: Direct-Access     Generic  STORAGE DEVICE   9228 PQ: 0
 ANSI: 0
[288955.717745] sd 6:0:0:0: Attached scsi generic sg4 type 0
[288961.876382] sd 6:0:0:0: [sdd] Attached SCSI removable disk
```

3. Use the **dd** utility to write the image. *Make sure you have the right drive!*

```
# dd if=/path/to/Fedora-Live-Security-x86_64-21.iso of=/dev/sdd
```

Procedure 3.6. Creating a Boot CD or DVD

In addition to creating a bootable USB flash drive, you can also use the provided ISO images to create bootable optical media (a CD or DVD). This approach may be necessary when installing Fedora on an older system which can not boot from USB.

[1] https://github.com/downloads/openSUSE/kiwi/ImageWriter.exe

[2] http://www.netbsd.org/~martin/rawrite32/

Note

The exact steps you need to take to burn a bootable CD or DVD from an ISO image will vary depending on what disc burning software you use. This procedure only offers a general overview.

1. Insert a blank CD or DVD into your system's CD or DVD burner.
2. Open your system's burning software - for example, **Brasero** on Fedora systems with **GNOME** desktop environment, or **Nero** on Windows systems. In the software's main menu, find an option which lets you burn an ISO image to a disc. For example, in **Brasero**, this option is **Burn image** in the main menu on the left side of the window.
3. When prompted, select the ISO image of Fedora to be burned, and the CD or DVD burner with a blank disc inside (if you have more than one drive).
4. Confirm your selection, and wait for the disc to be burned.

Chapter 4.

Booting the Installation

This chapter will explain how to boot the installer from local media or a network server as well as how to navigate the boot menu and use custom options.

4.1. Preparing to Boot

After you have made a bootable USB flash drive or a CD or DVD using the steps described in *Section 3.4, “Preparing Boot Media”*, you are ready to boot the installation. Note that the steps described below are generic and the exact steps will vary somewhat depending on your system - particularly on your motherboard manufacturer.

Warning

There are no separate media provided for BIOS and UEFI systems; all of them can boot from the same ISO image. However, once you install Fedora, you can not switch between UEFI and BIOS. The system must run on the same firmware it was installed on; if you, for example, perform the installation on an UEFI system in UEFI mode, and then switch it to BIOS compatibility mode, Fedora will no longer boot and will require a reinstallation.

Important

Fedora does not support UEFI booting for 32-bit x86 systems. Only BIOS boot is supported on these systems.

Also note that Fedora only fully supports version 2.2 of the UEFI specification. Hardware that supports version 2.3 or later will boot and work normally, but the additional functionality defined by these later specifications will not be available. The UEFI specifications are available from *http://www.uefi.org/specs/agreement/*.

To boot the Fedora installer, follow these steps:

Procedure 4.1. Booting the Fedora Installer

1. Plug in the boot USB drive, or insert the boot CD or DVD into your computer's optical disc drive. Alternatively, if you plan on booting from a network boot (PXE) server, make sure that the network cable is plugged in.

2. Restart the system. Once it starts rebooting, it should display a prompt similar to the following (usually at the bottom of the screen):

   ```
   Press F12 to select boot device, or Del to enter SETUP
   ```

   Follow the on-screen instructions to access the boot menu. If no instructions are displayed (some systems only display a graphical logo during early stages of boot), try pressing **F12**, **F11**, **F10** or **Del** several times; these are most commonly used keys. Note that there is usually a very short time window provided to access the menu; once it passes, you need to restart the system and try again.

Note

Some older systems may not support choosing a boot device at startup. In that case, enter the system's SETUP (BIOS), and change the default boot order so that your boot media (CD, DVD, USB or network) have higher priority than internal hard drives.

3. When your system's boot menu opens, select an entry such as **Boot from USB** if you created a bootable USB drive, **Boot from CD/DVD** if you are using an optical disc to install Fedora, or **Boot from PXE** if you want to boot from a network location.

4. Wait until the boot menu is displayed. The boot menu is described further in this chapter.

4.2. The Boot Menu

In most cases, when you boot the Fedora installer from your prepared boot media or server, the boot menu will the the first thing that appears. From this menu, you can either start the actual installation, or you can use the boot media to rescue an existing system.

The way the boot menu will look and function will vary somewhat depending on your system's firmware - BIOS systems use the **SYSLINUX** boot loader, and UEFI systems use **GRUB2**. However, both of the menus described below function very similarly from a user's point of view.

Use arrow keys to select an entry in the menu, and **`Enter`** to confirm your selection. The first two entries in the list will both proceed with the installation; the first one will start the installer directly, and the second one will verify the integrity of the boot media before starting the installation.

The final entry in the list is **Troubleshooting**; this is a submenu. Selecting this entry and pressing **`Enter`** will display a new set of selections, where you can choose to install in basic graphics mode (useful if you want to do a manual graphical installation but your system has issues with the default graphical installer), rescue an existing system, or test your system's memory modules for errors (on BIOS systems only). The troubleshooting menu also allows you to exit the boot menu and boot normally from your system's hard drive via the **Boot from local drive** option.

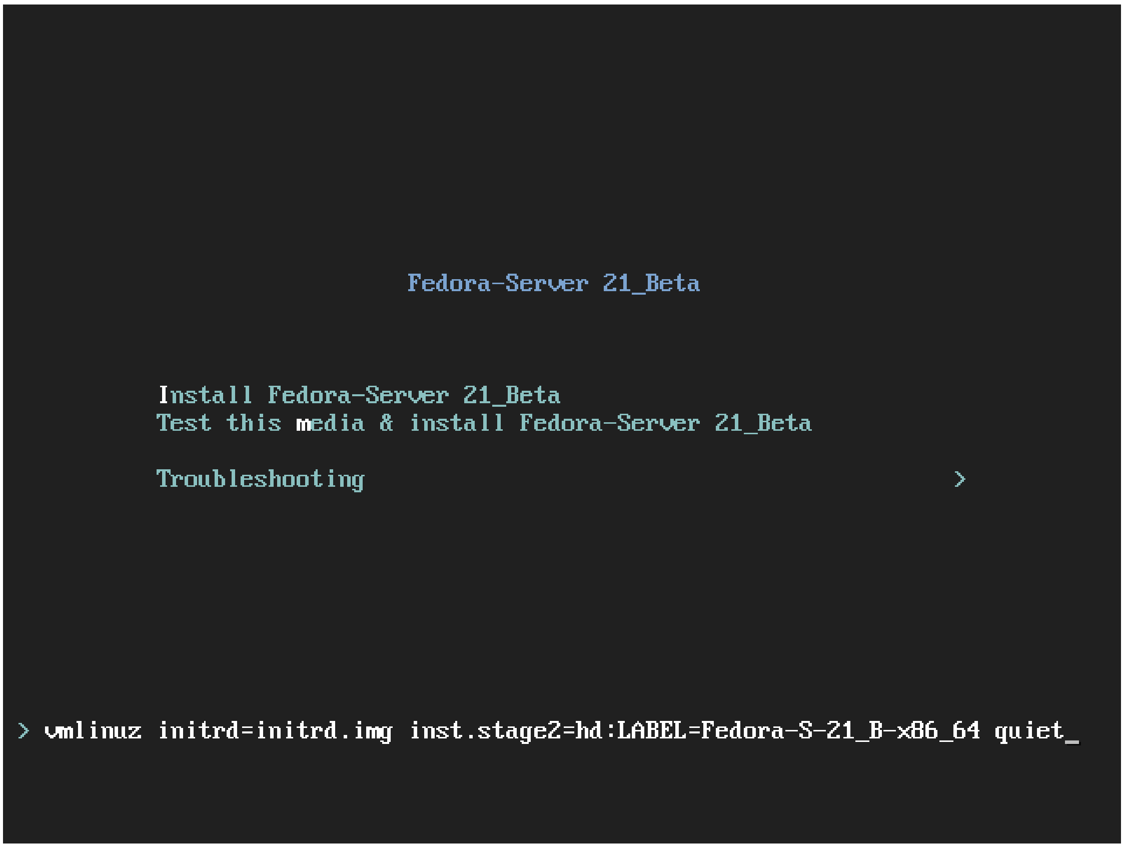
Every menu entry in the list is a predefined set of boot options, and these options can be customized to change some aspects of the installer's behavior. To edit the default set of boot options, press **`Tab`** on BIOS systems, or **`e`** on UEFI systems. The key to use is also displayed at the bottom of the screen.

Note

Editing boot options in **GRUB2** (on UEFI systems) will display the entire configuration for the selected entry. Actual boot options are configured on the line which starts with the **`linux`** (or **`linux16`** or **`linuxefi`**) keyword. Do not modify any other lines in the configuration.

On systems with BIOS firmware, only the actual boot options are displayed when you press **`Tab`**.

When editing the default set of options, you can change the existing ones as well as append additional ones. Once you finish, press **`Enter`** on BIOS or **`Ctrl`**+**`X`** on UEFI to boot the installer using your customized options.


Fedora-Server 21_Beta
Install Fedora-Server 21_Beta
Test this media & install Fedora-Server 21_Beta
Troubleshooting >

```
> vmlinuz initrd=initrd.img inst.stage2=hd:LABEL=Fedora-S-21_B-x86_64 quiet_
```

Figure 4.1. Editing boot options on a system with BIOS firmware

All available **Anaconda** boot options are described in *Section 8.2, “Available Boot Options”*.

Installing Using Anaconda

This chapter provides step-by-step instructions for installing Fedora using the **Anaconda** installer. The bulk of this chapter describes installation using the graphical user interface. A text mode is also available for systems with no graphical display, but this mode is limited in certain aspects (for example, custom partitioning is not possible in text mode).

If your system does not have the ability to use the graphical mode, you can:

- Use Kickstart to automate the installation as described in *Chapter 9, Automating the Installation with Kickstart*
- Perform the graphical installation remotely by connecting to the installation system from another computer with a graphical display using the *VNC* (Virtual Network Computing) protocol - see *Chapter 11, Installing Using VNC*

5.1. Introduction to Anaconda

The Fedora installer, **Anaconda**, is different from most other operating system installation programs due to its parallel nature. Most installers follow a fixed path: you must choose your language first, then you configure network, then installation type, then partitioning, etc. There is usually only one way to proceed at any given time.

In **Anaconda** you are only required to select your language and locale first, and then you are presented with a central screen, where you can configure most aspects of the installation in any order you like. This does not apply to all parts of the installation process, however - for example, when installing from a network location, you must configure the network before you can select which packages to install.

Some screens will be automatically configured depending on your hardware and the type of media you used to start the installation. You can still change the detected settings in any screen. Screens which have not been automatically configured, and therefore require your attention before you begin the installation, are marked by an exclamation mark. You can not start the actual installation process before you finish configuring these settings.

Additional differences appear in certain screens; notably the custom partitioning process is very different from other Linux distributions. These differences are described in each screen's subsection.

5.2. Consoles and Logging During the Installation

The following sections describe how to access logs and an interactive shell during the installation. This is useful when troubleshooting problems, but should not be necessary in most cases.

5.2.1. Accessing Consoles

The Fedora installer uses the **tmux** terminal multiplexer to display and control several windows you can use in addition to the main interface. Each of these windows serves a different purpose - they display several different logs, which can be used to troubleshoot any issues during the installation, and one of the windows provides an interactive shell prompt with `root` privileges, unless this prompt was specifically disabled using a boot option or a Kickstart command.

Note

In general, there is no reason to leave the default graphical installation environment unless you need to diagnose an installation problem.

The terminal multiplexer is running in virtual console 1. To switch from the actual installation environment to **tmux**, press **Ctrl**+**Alt**+**F1**. To go back to the main installation interface which runs in virtual console 6, press **Ctrl**+**Alt**+**F6**.

Note

If you choose text mode installation, you will start in virtual console 1 (**tmux**), and switching to console 6 will open a shell prompt instead of a graphical interface.

The console running **tmux** has 5 available windows; their contents are described in the table below, along with keyboard shortcuts used to access them. Note that the keyboard shortcuts are two-part: first press **Ctrl**+**b**, then release both keys, and press the number key for the window you want to use.

You can also use **Ctrl**+**b** **n** and **Ctrl**+**b** **p** to switch to the next or previous **tmux** window, respectively.

Table 5.1. Available tmux Windows

Shortcut	Contents
Ctrl+**b** **1**	Main installation program window. Contains text-based prompts (during text mode installation or if you use VNC direct mode), and also some debugging information.
Ctrl+**b** **2**	Interactive shell prompt with `root` privileges.
Ctrl+**b** **3**	Installation log; displays messages stored in **/tmp/anaconda.log**.
Ctrl+**b** **4**	Storage log; displays messages related storage devices from kernel and system services, stored in **/tmp/storage.log**.
Ctrl+**b** **5**	Program log; displays messages from other system utilities, stored in **/tmp/program.log**.

In addition to displaying diagnostic information in **tmux** windows, **Anaconda** also generates several log files, which can be transferred from the installation system. These log files are described in *Section 7.1.1, "Log Files Generated During the Installation"*, and directions for transferring them from the installation system are available in *Section 7.1.2, "Transferring Log Files from the Installation System"*.

5.2.2. Saving Screenshots

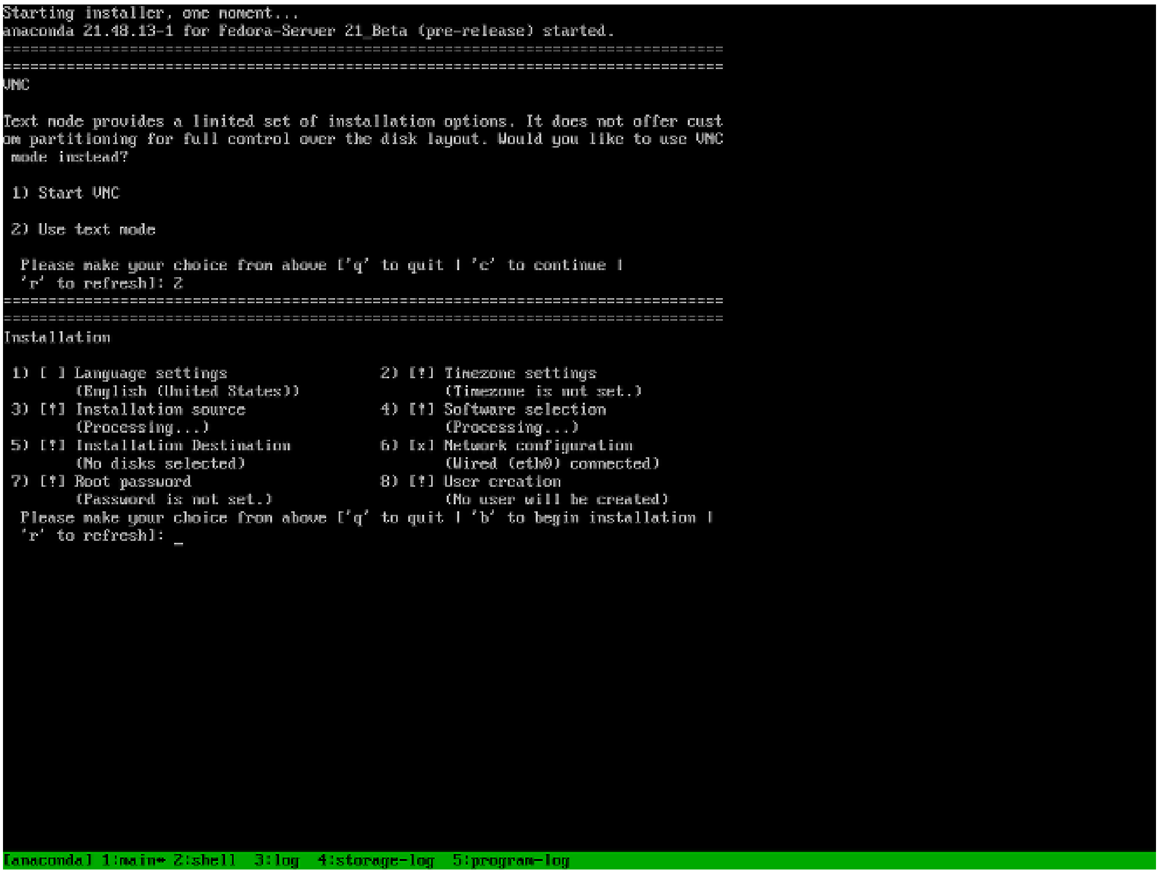
You can press **Shift**+**Print Screen** at any time during the graphical installation to capture the current screen. These screenshots are saved to **/tmp/anaconda-screenshots**.

Additionally, you can use the **autostep --autoscreenshot** command in a Kickstart file to capture and save each step of the installation automatically. See *Section A.6.1, "autostep (optional) - Go Through Every Screen"* for details.

5.3. Installing in Text Mode

Text mode installation offers an interactive, non-graphical interface for installing Fedora. This may be useful on systems with no graphical capabilities; however, you should always consider the available alternatives before starting a text-based installation. Text mode is limited in the amount of choices you can make during the installation.

There are two alternatives to text mode which can both be used even if the installation system does not have a graphical display. You can either connect to the installation system using VNC and perform an interactive graphical installation remotely (see *Chapter 11, Installing Using VNC*), or you can create a Kickstart file to perform the installation automatically (see *Chapter 9, Automating the Installation with Kickstart*).



```
Starting installer, one moment...
anaconda 21.48.13-1 for Fedora-Server 21_Beta (pre-release) started.
================================================================================
================================================================================
VNC

Text mode provides a limited set of installation options. It does not offer cust
om partitioning for full control over the disk layout. Would you like to use VNC
 mode instead?

 1) Start VNC

 2) Use text mode

  Please make your choice from above ['q' to quit | 'c' to continue |
  'r' to refresh]: 2
================================================================================
================================================================================
Installation

 1) [ ] Language settings                 2) [!] Timezone settings
        (English (United States))                (Timezone is not set.)
 3) [!] Installation source               4) [!] Software selection
        (Processing...)                          (Processing...)
 5) [!] Installation Destination          6) [x] Network configuration
        (No disks selected)                      (Wired (eth0) connected)
 7) [!] Root password                     8) [!] User creation
        (Password is not set.)                   (No user will be created)
  Please make your choice from above ['q' to quit | 'b' to begin installation |
  'r' to refresh]: _













[anaconda] 1:main* 2:shell  3:log  4:storage-log  5:program-log
```

Figure 5.1. Text Mode Installation

Installation in text mode follows a pattern similar to the graphical installation: There is no single fixed progression; you can configure many settings in any order you want using the main status screen. Screens which have already been configured, either automatically or by you, are marked as `[x]`, and screens which require your attention before the installation can begin are marked with `[!]`. Available commands are displayed below the list of available options.

Limits of interactive text mode installation include:

- The installer will always use the English language and the US English keyboard layout. You can configure your language and keyboard settings, but these settings will only apply to the installed system, not to the installation.

- You can not configure any advanced storage methods (LVM, software RAID, FCoE, zFCP and iSCSI).

- It is not possible to configure custom partitioning; you must use one of the automatic partitioning settings. You also cannot configure where the boot loader will be installed.

To start a text mode installation, boot the installation with the **`inst.text`** boot option used either at the boot command line in the boot menu, or in your PXE server configuration. See *Chapter 4, Booting the Installation* for information about booting and using boot options.

5.4. Installing in the Graphical User Interface

The graphical installation interface is the preferred method of manually installing Fedora. It allows you full control over all available settings, including custom partitioning and advanced storage configuration, and it is also localized to many languages other than English, allowing you to perform the entire installation in a different language. The graphical mode is used by default when you boot the system from local media (a CD, DVD or a USB flash drive).

The sections below discuss each screen available in the installation process. Note that due to the installer's parallel nature, most of the screens do not have to be completed in the order in which they are described here.

Each screen in the graphical interface contains a **Help** button. This button opens the **Yelp** help browser displaying the section of the [Fedora Installation Guide] relevant to the current screen.

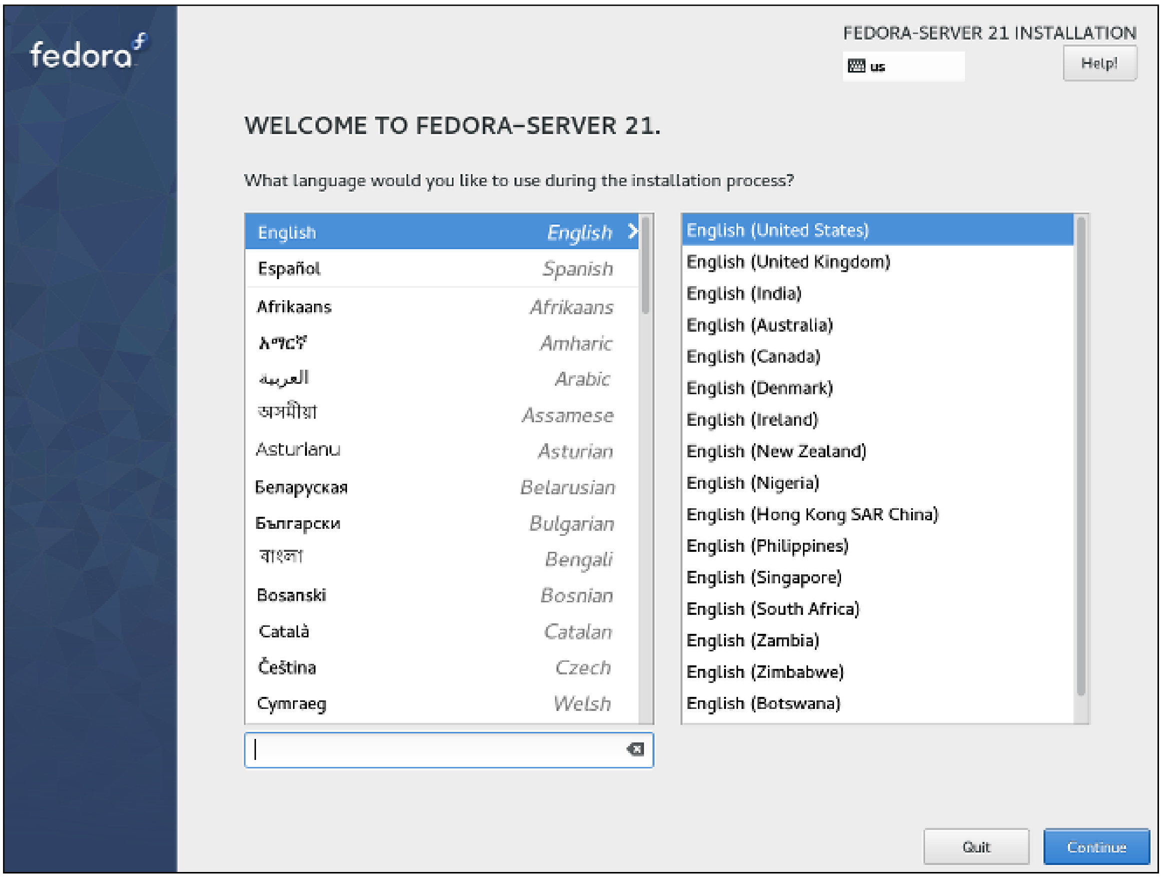
You can also control the graphical installer with your keyboard. Use **`Tab`** and **`Shift`**+**`Tab`** to cycle through active control elements (buttons, check boxes, etc.) on the current screen, **`Up`** and **`Down`** arrow keys to scroll through lists, and **`Left`** and **`Right`** to scroll through horizontal toolbars or table entries. **`Space`** or **`Enter`** can be used to select or remove a highlighted item from selection and to expand and collapse drop-down menus.

Additionally, elements in each screen can be toggled using their respective shortcuts. These shortcuts are highlighted (underlined) when you hold down the **`Alt`** key; to toggle that element, press **`Alt`**+***`X`***, where *X* is the highlighted letter.

Your current keyboard layout is displayed in the top right hand corner. Only one layout is configured by default; if you configure more than layout in the **Keyboard Layout** screen (*Section 5.4.4, “Keyboard Layout”*), you can switch between them by clicking the layout indicator.

5.4.1. Welcome Screen and Language Selection

The first screen displayed immediately after the graphical installer starts is the Welcome screen.



WELCOME TO FEDORA-SERVER 21.

What language would you like to use during the installation process?

English	English
Español	Spanish
Afrikaans	Afrikaans
አማርኛ	Amharic
العربية	Arabic
অসমীয়া	Assamese
Asturianu	Asturian
Беларуская	Belarusian
Български	Bulgarian
বাংলা	Bengali
Bosanski	Bosnian
Català	Catalan
Čeština	Czech
Cymraeg	Welsh

English (United States)
English (United Kingdom)
English (India)
English (Australia)
English (Canada)
English (Denmark)
English (Ireland)
English (New Zealand)
English (Nigeria)
English (Hong Kong SAR China)
English (Philippines)
English (Singapore)
English (South Africa)
English (Zambia)
English (Zimbabwe)
English (Botswana)

Figure 5.2. Welcome Screen

First, select your preferred language in the left hand side column, and then select your locale from the right hand side column. You can use the text input field in the bottom left corner to search for your language instead of looking for it in the full list of more than 70 languages.

The language you select on this screen will be used during the installation, and it will also be used on the installed system by default. You can change the language for the installed system later, but once you click **Continue** on this screen, you will not be able to go back and change the language used inside the installer itself.

One language is pre-selected by default on top of the list. If network access is configured at this point (for example, if you booted from a network server instead of local media), the pre-selected language will be determined based on automatic location detection using the GeoIP module. Alternatively, if you used the `inst.lang=` option on the boot command line or in your PXE server configuration, this language will be selected by default, but you will still be able to change it.

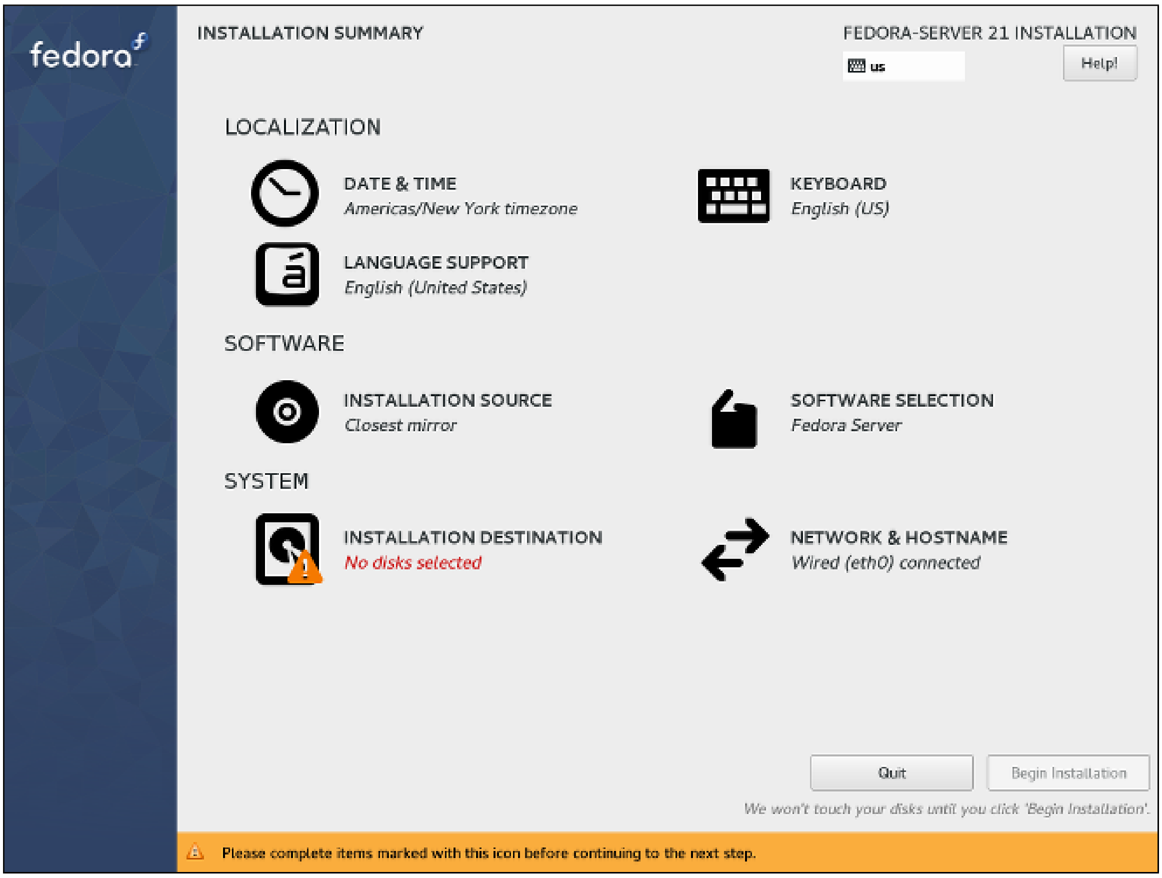
After you select your language and locale, click **Continue** to confirm your selection and proceed to *Section 5.4.2, “Installation Summary”*.

Note

If you are installing a pre-release version of Fedora, a message will be shown after you click **Continue** warning you about the pre-release status of your installation media. Click **I accept my fate** to continue with the installation, or **Get me out of here** to quit the installation and reboot your system.

5.4.2. Installation Summary

The **Installation Summary** screen is the central location for setting up an installation. Most of the options which can be configured during the installation can be accessed from here.



INSTALLATION SUMMARY

FEDORA-SERVER 21 INSTALLATION

us

Help!

LOCALIZATION

DATE & TIME
Americas/New York timezone

KEYBOARD
English (US)

LANGUAGE SUPPORT
English (United States)

SOFTWARE

INSTALLATION SOURCE
Closest mirror

SOFTWARE SELECTION
Fedora Server

SYSTEM

INSTALLATION DESTINATION
No disks selected

NETWORK & HOSTNAME
Wired (eth0) connected

Quit

Begin Installation

We won't touch your disks until you click 'Begin Installation'.

⚠ Please complete items marked with this icon before continuing to the next step.

Figure 5.3. Installation Summary

Note

If you used a Kickstart option or a boot option to specify an installation repository on a network, but no network is available at the start of the installation, the installer will display the **Network Configuration** screen for you to set up a network connection prior to displaying the **Installation Summary** screen.

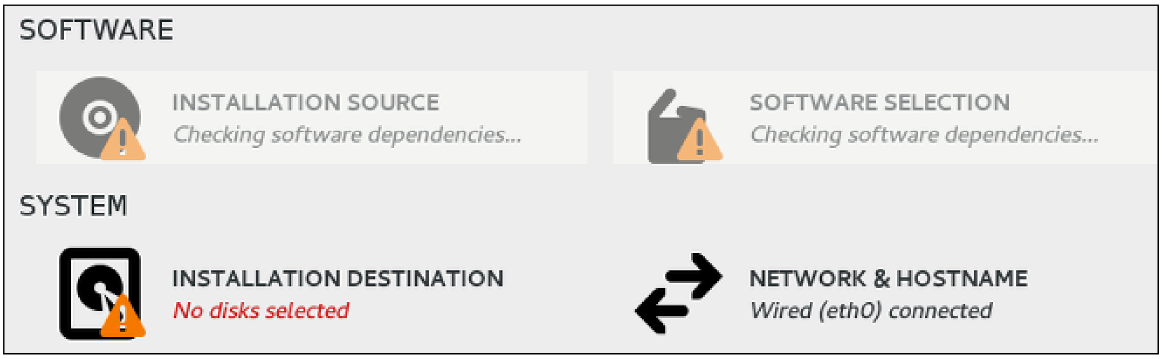
The summary screen consists of several links to other screens, separated into categories. These links can be in several different states, which are graphically indicated:

- A *warning symbol* (yellow triangle with an exclamation mark) next to an icon means that a screen requires your attention before you start the installation. This typically happens with the **Installation Destination** screen, because even though there is a default automatic partitioning variant, you always have to at least confirm this selection, even if you do not want to make any changes.

- If a link is *greyed out*, it means that the installer is currently configuring this section, and you must wait for the configuration to finish before accessing that screen. This typically happens when you change the installation source in the **Installation Source** screen and the installer is probing the new source location and gathering a list of available packages.

- Screens with *black text and no warning symbol* mean that this screen does not require your attention. You can still change your settings in these screens, but it is not necessary to do so to complete the installation. This typically happens with localization settings, as these are either detected automatically, or set up on the previous screen where you select your language and locale.

A warning message is displayed at the bottom of the summary screen, and the **Begin Installation** button is greyed out, as long as at least one item has not been configured yet.



SOFTWARE

INSTALLATION SOURCE
Checking software dependencies...

SOFTWARE SELECTION
Checking software dependencies...

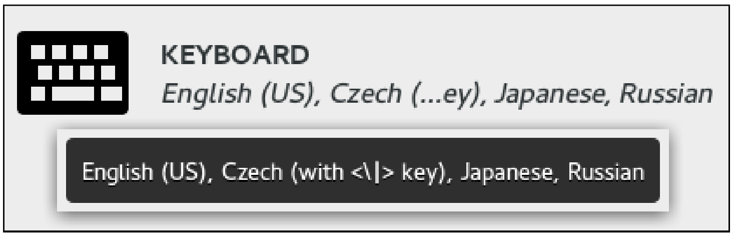
SYSTEM

INSTALLATION DESTINATION
No disks selected

NETWORK & HOSTNAME
Wired (eth0) connected

Figure 5.4. Icon States in Installation Summary

Each screen also has an explanatory text below its title, showing settings currently configured in that screen. This text may be concatenated; in that case, move your mouse cursor over it and wait until a tooltip with the full text appears.



KEYBOARD
English (US), Czech (...ey), Japanese, Russian

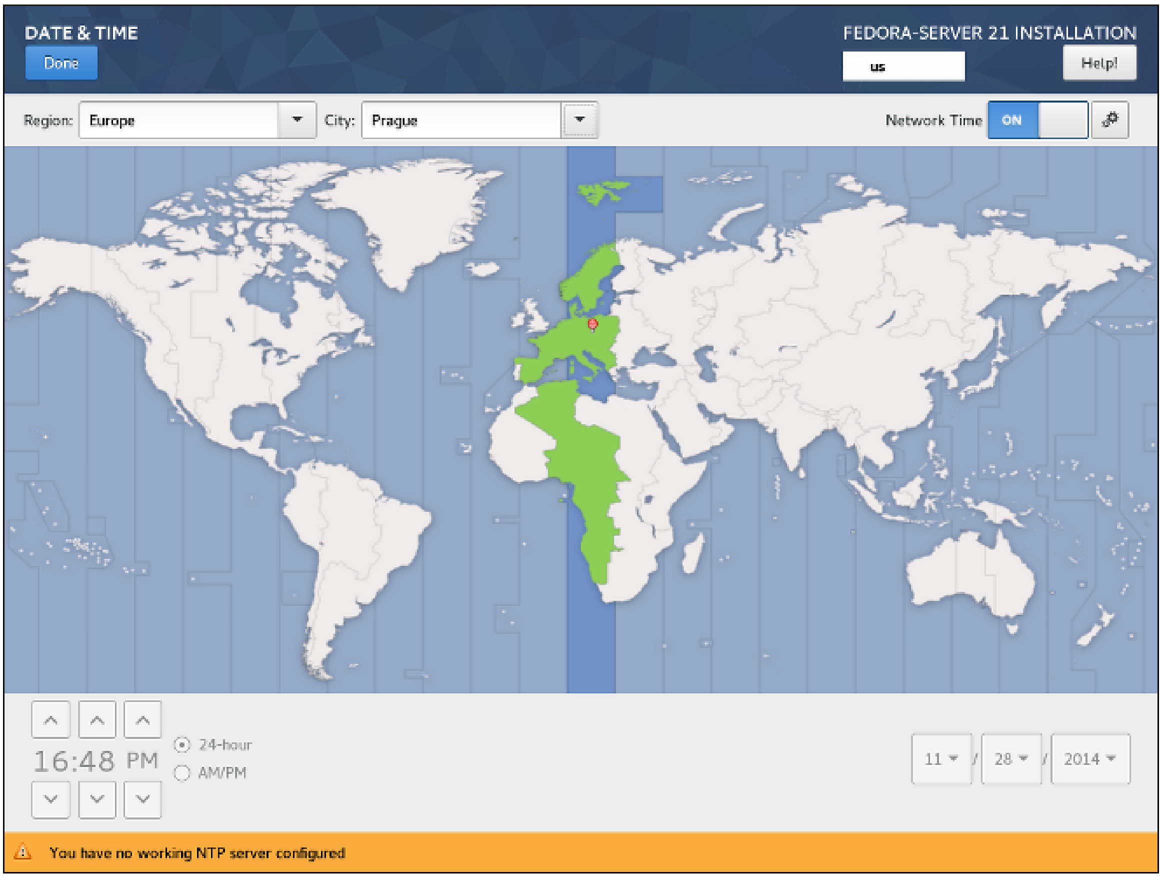
English (US), Czech (with <\|> key), Japanese, Russian

Figure 5.5. Tooltip in the Installation Summary

Once you configure everything required for the installation, you can press the **Begin Installation** button to start installing Fedora. This will take you to *Section 5.4.13, “Configuration and Installation Progress”*. Note that as the text below this button says, nothing will be written to your hard drive before you press this button. You can press **Quit** at any point; this will discard all changes you made in the installer so far and reboot the system.

5.4.3. Date & Time

The **Date & Time** screen allows you to configure time and date-related settings for your system. This screen is automatically configured based on the settings you selected in *Section 5.4.1, “Welcome Screen and Language Selection”*, but you can change your date, time and location settings before you begin the installation.



us

Region: Europe

City: Prague

Network Time ON

16:48 PM

● 24-hour

○ AM/PM

11 / 28 / 2014

You have no working NTP server configured

Figure 5.6. Date & Time

First, select your **Region** using the drop-down menu in the top left corner of the screen. Then, select your **City**, or the city closest to your location in the same time zone. Selecting a specific location helps Fedora ensure that your time is always set correctly including automatic time changes for daylight savings time if applicable.

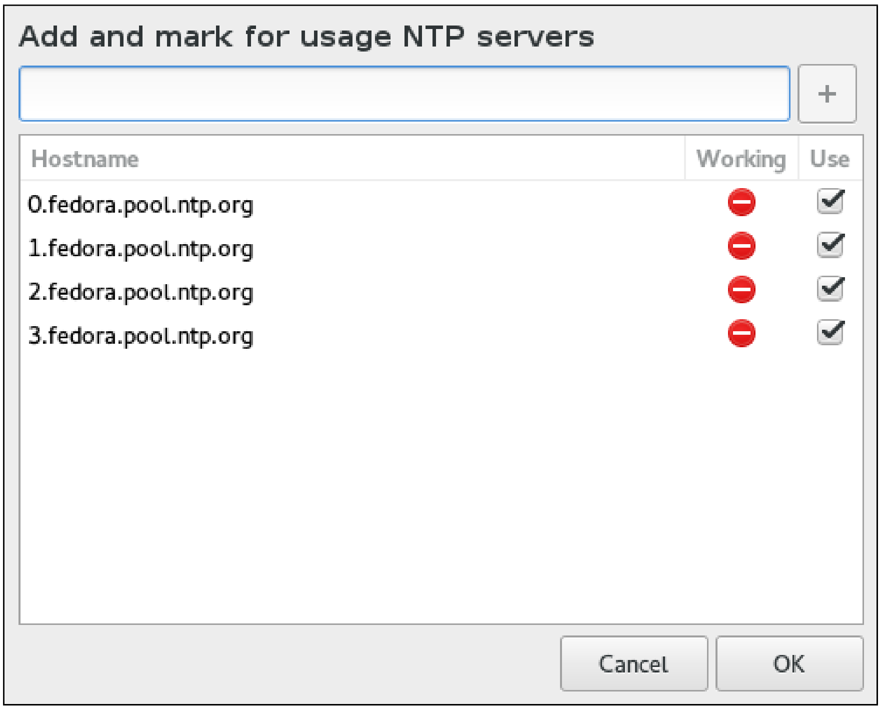
You can also select a time zone relative to Greenwich Mean Time (GMT) without setting your location to a specific region. To do so, select **Etc** as your region.

Note

The list of cities and regions comes from the Time Zone Database (*tzdata*) public domain, which is maintained by the Internet Assigned Numbers Authority (IANA). The Fedora Project can not add cities or regions into this database. You can find more information at the *IANA official website*[1].

The switch labeled **Network Time** in the top right corner of the screen can be used to enable or disable network time synchronization using the Network Time Protol (NTP). Enabling this option will keep your system time correct as long as the system can access the internet. By default, four NTP *pools* are configured; you can add others and disable or remove the default ones by clicking the gear wheel button next to the switch.

[1] http://www.iana.org/time-zones



Hostname	Working	Use
0.fedora.pool.ntp.org	⛔	☑
1.fedora.pool.ntp.org	⛔	☑
2.fedora.pool.ntp.org	⛔	☑
3.fedora.pool.ntp.org	⛔	☑

Figure 5.7. The Add and mark for usage NTP servers dialog

If you disable network time synchronization, the controls at the bottom of the screen will become active, and you will be able to set the current time and date manually.

After configuring your time and date settings, press the **Done** button in the top left corner to return to *Section 5.4.2, “Installation Summary”*.

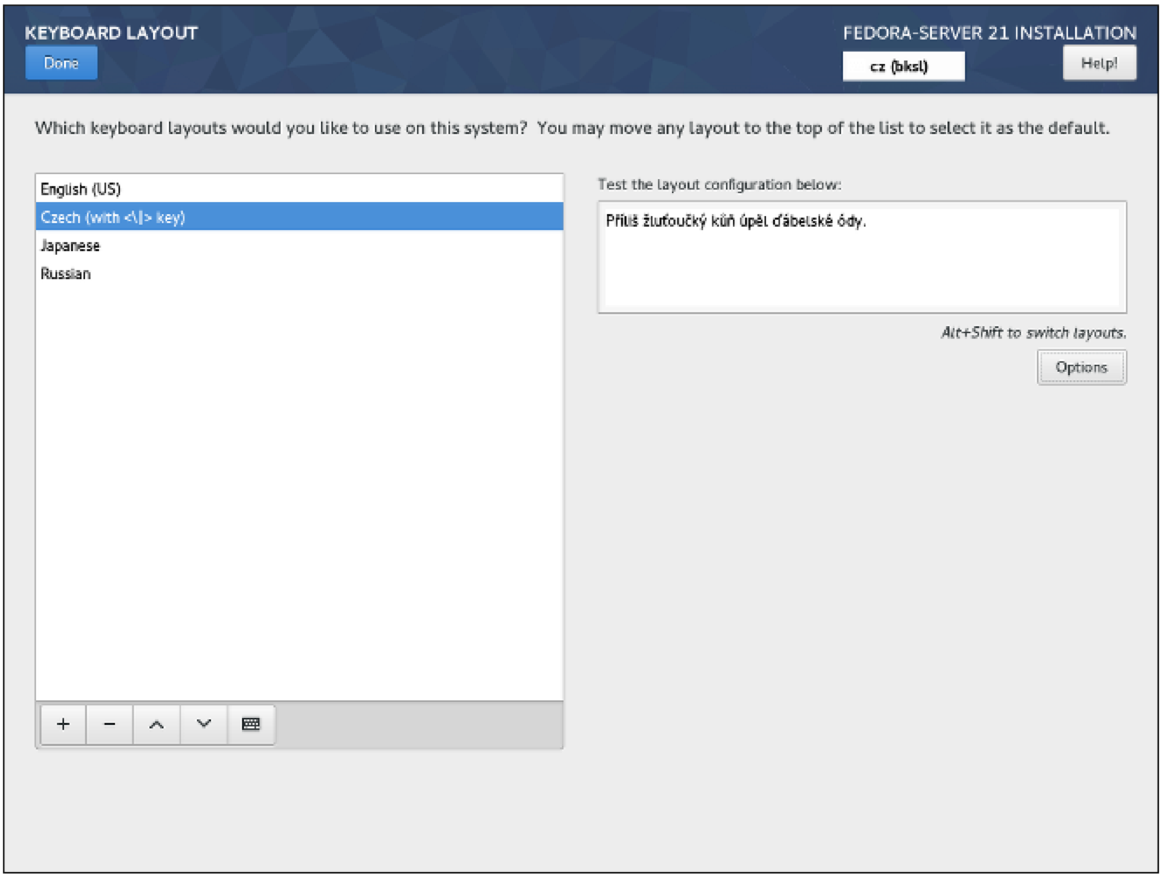
5.4.4. Keyboard Layout

The **Keyboard Layout** screen allows you to set up one or more keyboard layouts for your system and a way to switch between them. One keyboard layout is configured automatically based on your selection in *Section 5.4.1, “Welcome Screen and Language Selection”*, but you can change this layout and add additional ones before you begin the installation.

Keyboard layouts are a separate setting from system languages, and these two settings can be mixed as you see fit.

Note

All settings configured in this screen will be available on the installed system, and they will also become immediately available inside the installer. You can use the keyboard icon in the top right corner of any screen, or the keyboard switch you configured in this screen, to cycle between your configured layouts.



Which keyboard layouts would you like to use on this system? You may move any layout to the top of the list to select it as the default.

- English (US)
- Czech (with <\|> key)
- Japanese
- Russian

Test the layout configuration below:

Příliš žluťoučký kůň úpěl ďábelské ódy.

Alt+Shift to switch layouts.

Figure 5.8. Keyboard Layout

The right half of the screen contains a window listing all currently configured layouts. The order in which the layouts are displayed is important - the same order will be used when switching between layouts, and the first listed layout will be the default on your system.

The text field on the right side of the screen can be used to test the currently selected layout.

You can click a layout in the list to highlight it. At the bottom of the list, there is a set of buttons:

- The **+** button adds a new layout. When you press this button, a new window opens with a list of all available layouts, grouped by language. You can find a layout by browsing the list, or you can use the search bar at the bottom of this window. When you find the layout you want to add, highlight it and press **Add**.
- The **-** button removes the currently highlighted layout.
- The up and down buttons can be used to move the highlighted layout up or down in the list.
- The keyboard button opens a new window which offers a visual representation of the highlighted layout.

Important

If you use a layout that cannot accept Latin characters, such as **Russian**, you are advised to also add the **English (United States)** layout and configure a keyboard combination to switch between the two layouts. If you only select a layout without Latin characters, you may be unable to enter a valid `root` password and user credentials later in the installation process. This may prevent you from completing the installation.

You can also optionally configure a keyboard switch which can be used to cycle between available layouts. To do so, click the **Options** button on the right side of the screen. The **Layout Switching Options** dialog will open, allowing you to configure one or more keys or key combinations for switching. Select one or more key combinations using the check boxes next to them, and click **OK** to confirm your selection.

After you finish configuring keyboard layouts and switches, click **Done** in the top left corner to return to *Section 5.4.2, “Installation Summary”*.

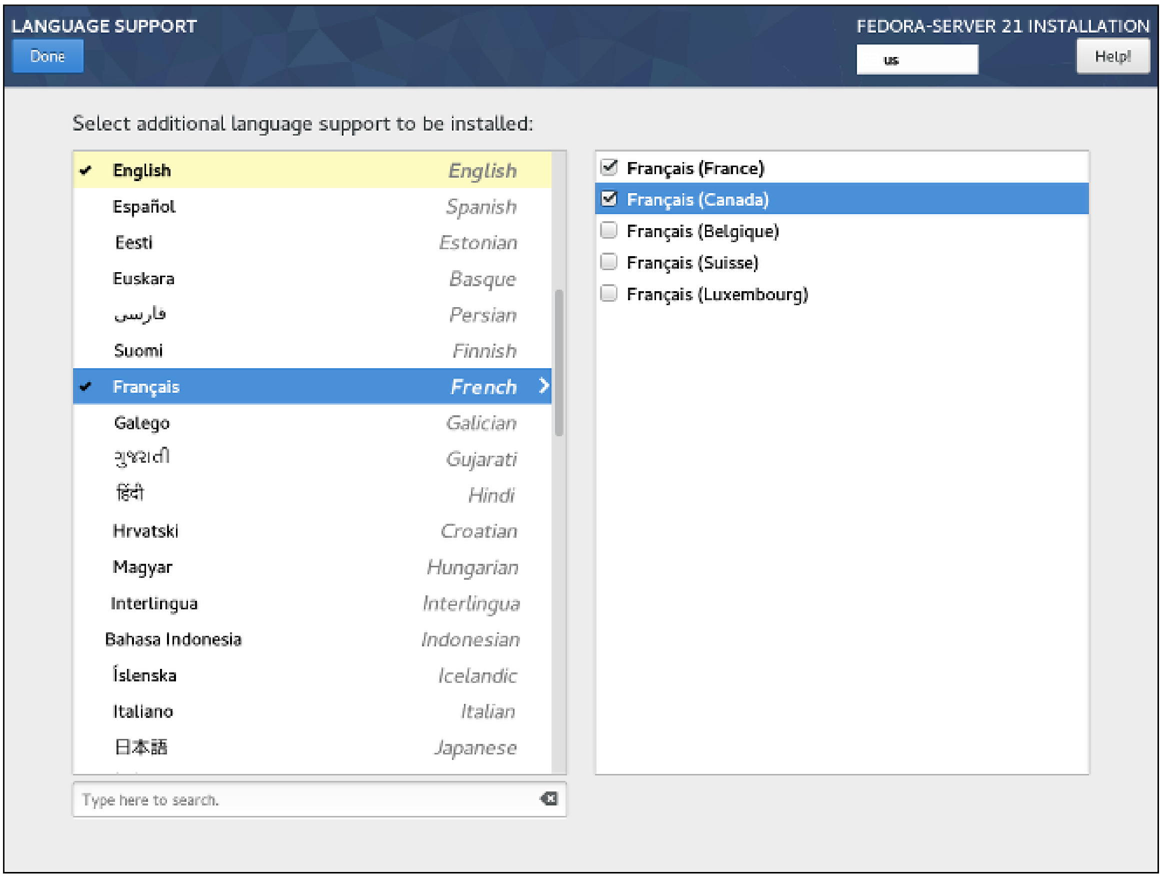
5.4.5. Language Support

The **Language Support** screen allows you to configure language settings for your system. The default language is determined by your selection in *Section 5.4.1, “Welcome Screen and Language Selection”* and support for this language can not be removed. You can only add additional languages, which will be available on the installed system - not during the installation.

If you want to change the default language, or the language used during the installation, you must reboot your system, start the installer again, and select a different language in *Section 5.4.1, “Welcome Screen and Language Selection”*.

Note

Adding support for another language does not automatically configure the corresponding keyboard layout. Layouts are a separate setting configured in *Section 5.4.4, “Keyboard Layout”*.



Select additional language support to be installed:

✓	English	English
	Español	Spanish
	Eesti	Estonian
	Euskara	Basque
	فارسی	Persian
	Suomi	Finnish
✓	Français	French
	Galego	Galician
	ગુજરાતી	Gujarati
	हिंदी	Hindi
	Hrvatski	Croatian
	Magyar	Hungarian
	Interlingua	Interlingua
	Bahasa Indonesia	Indonesian
	Íslenska	Icelandic
	Italiano	Italian
	日本語	Japanese

☑ Français (France)
☑ Français (Canada)
☐ Français (Belgique)
☐ Français (Suisse)
☐ Français (Luxembourg)

Figure 5.9. Language Support

The left panel contains a list of available language groups such as **English** or **Bulgarian**. If at least one language from a group is selected, a check mark will be displayed next to the group, and the list entry will be highlighted. This allows you to easily see which languages you have configured support for.

To add support for one or more additional languages, click a group in the left panel, and then select one or more regional variations in the right panel using check boxes next to list entries. Repeat this process for all languages you want to install support for.

Note

Enabling support for some languages (typically languages which use non-Latin script) will install additional packages - for example, enabling support for one or more languages from the **Arabic** group will also install the `arabic-support` package group. For more information about packages, see *Section 5.4.7, “Software Selection”*.

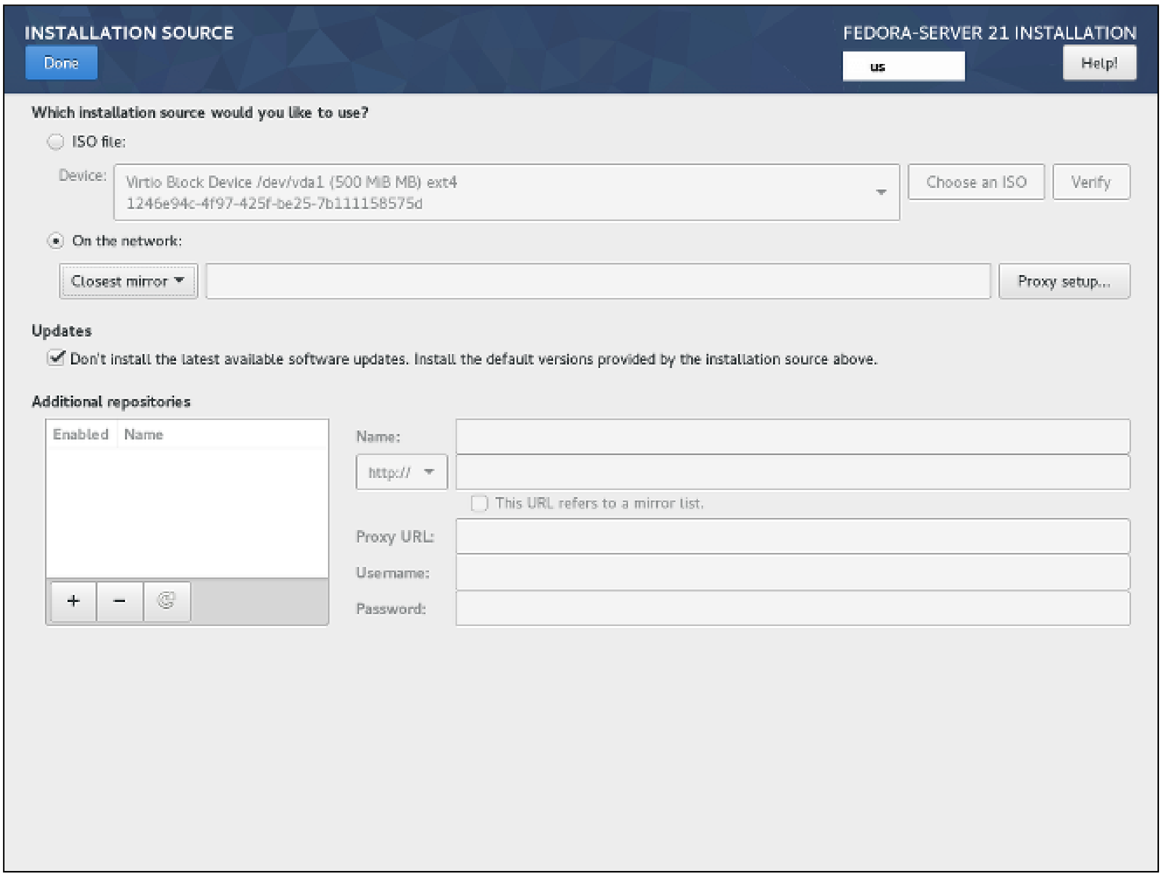
Once you have made your selections, click **Done** in the top left corner to return to *Section 5.4.2, “Installation Summary”*.

5.4.6. Installation Source

The **Installation Source** screen allows you to specify a location (local or on the network) from which packages will be downloaded and installed on your system. This screen will be configured automatically in most cases, but you can change your settings or add additional sources.

Note

Normally, when you first enter the **Installation Summary** screen, the installer will attempt to configure an installation source based on the type of media you used to boot. The full Fedora Server DVD will configure the source as local media, the netinst ISO image will configure the closest network mirror, etc. This process takes some time, especially if the default source is a network mirror. If you plan to use a custom installation source, use the **`inst.askmethod`** boot option to skip the initial configuration; this will allow you to enter this screen immediately. See *Section 8.2.1, “Specifying the Installation Source”* for information about boot options.



Which installation source would you like to use?

○ ISO file:

Device: Virtio Block Device /dev/vda1 (500 MiB MB) ext4 1246e94c-4f97-425f-be25-7b111158575d

● On the network:

Closest mirror

Updates

☑ Don't install the latest available software updates. Install the default versions provided by the installation source above.

Additional repositories

Enabled	Name

Name:

http://

□ This URL refers to a mirror list.

Proxy URL:

Username:

Password:

Figure 5.10. Installation Source

The following options are available. Note that not all of them may be displayed.

Auto-detected installation media
: This is the option selected by default if you started the installer from media containing an installation source, such as a live DVD. No additional configuration is necessary. You can click the **Verify** button check the media integrity.

ISO file
: This option will appear if the installation program detected a partitioned hard drive with mountable file systems during boot. Select this option, click the **Choose an ISO** button, and browse to the installation ISO file's location on your system. You can click the **Verify** button to check the file's integrity.

On the network

Use this option to download packages to be installed from a network location instead of local media. This is the default selection on network installation media.

In most cases, the **Closest mirror** option available from the protocol selection drop-down menu is preferable. If this option is selected, packages for your system will be downloaded from the most suitable location (mirror).

To manually configure a network-based installation source, use the drop-down menu to specify the protocol to be used when downloading packages. This setting depends on the server you want to use. Then, type the server address (without the protocol) into the address field. If you choose NFS, a second input field will appear where you can specify custom **NFS mount options**.

Important

When selecting an NFS installation source, you must specify the address with a colon (**:**) character separating the host name from the path. For example:

```
server.example.com:/path/to/directory
```

To configure a proxy for an HTTP or HTTPS source, click the **Proxy setup** button. Check **Enable HTTP proxy** and type the URL into the **Proxy URL** box. If the proxy server requires authentication, check **Use Authentication** and enter your user name and password. Click **OK** to finish the configuration.

If your HTTP or HTTPS URL refers to a repository mirror list, mark the check box under the address field.

You can also specify additional repositories in the **Additional repositories** section to gain access to more installation environments and software add-ons. All environments and add-ons will be available for selection in *Section 5.4.7, “Software Selection”* once you finish configuring the sources.

To add a repository, click the **+** button. To delete a repository, select one in the list and click the **-** button. Click the arrow icon to revert to the previous list of repositories, i.e. to replace current entries with those that were present at the time you entered the **Installation Source** screen. To activate or deactivate a repository, click the check box in the **Enabled** column at each entry in the list.

You can name your additional repository and configure it the same way as the primary repository on the network using the input fields on the right side of the section.

Once you have selected your installation source, click **Done** in the top left corner to return to *Section 5.4.2, “Installation Summary”*.

5.4.7. Software Selection

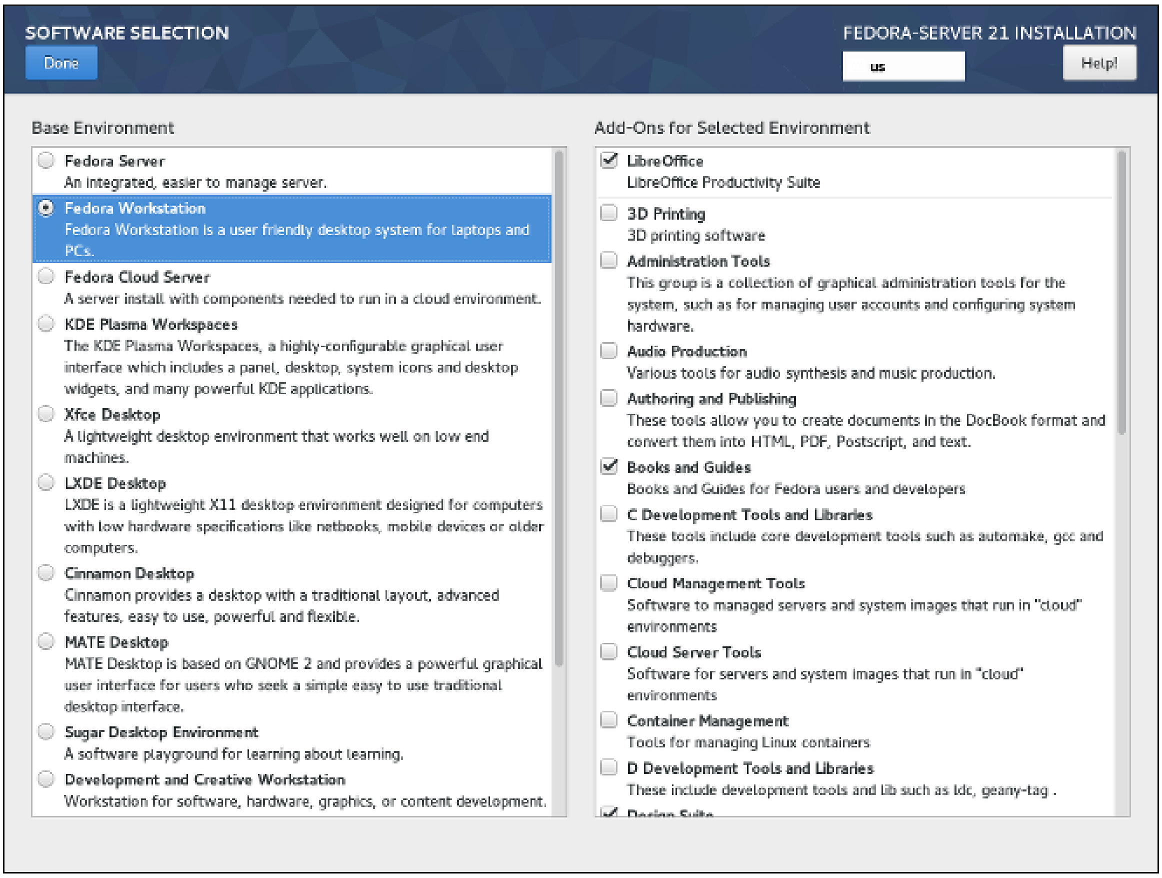
The **Software Selection** screen allows you to choose a *Base Environment* and *Add-ons*. These options control which software packages will be installed on your system during the installation process.

This screen is only available if *Section 5.4.6, “Installation Source”* is properly configured and only after the installer has downloaded package metadata from the source.

Note

It is not possible to select specific packages during a manual installation. You can only select pre-defined environments and add-ons. If you need to control exactly which packages are installed, you must use a Kickstart file and define the packages in the `%packages` section. See *Chapter 9, Automating the Installation with Kickstart* for information about Kickstart installations.

The availability of environments and add-ons depends on your installation source. By default, the selection depends on the installation media you used to start the installation; Fedora Server installation image will have different environments and add-ons available for selection than, for example, the Fedora Cloud image. You can change this by configuring a different installation source containing different environments.



Base Environment

- ○ Fedora Server
  An integrated, easier to manage server.
- ● Fedora Workstation
  Fedora Workstation is a user friendly desktop system for laptops and PCs.
- ○ Fedora Cloud Server
  A server install with components needed to run in a cloud environment.
- ○ KDE Plasma Workspaces
  The KDE Plasma Workspaces, a highly-configurable graphical user interface which includes a panel, desktop, system icons and desktop widgets, and many powerful KDE applications.
- ○ Xfce Desktop
  A lightweight desktop environment that works well on low end machines.
- ○ LXDE Desktop
  LXDE is a lightweight X11 desktop environment designed for computers with low hardware specifications like netbooks, mobile devices or older computers.
- ○ Cinnamon Desktop
  Cinnamon provides a desktop with a traditional layout, advanced features, easy to use, powerful and flexible.
- ○ MATE Desktop
  MATE Desktop is based on GNOME 2 and provides a powerful graphical user interface for users who seek a simple easy to use traditional desktop interface.
- ○ Sugar Desktop Environment
  A software playground for learning about learning.
- ○ Development and Creative Workstation
  Workstation for software, hardware, graphics, or content development.

Add-Ons for Selected Environment

- ☑ LibreOffice
  LibreOffice Productivity Suite
- □ 3D Printing
  3D printing software
- □ Administration Tools
  This group is a collection of graphical administration tools for the system, such as for managing user accounts and configuring system hardware.
- □ Audio Production
  Various tools for audio synthesis and music production.
- □ Authoring and Publishing
  These tools allow you to create documents in the DocBook format and convert them into HTML, PDF, Postscript, and text.
- ☑ Books and Guides
  Books and Guides for Fedora users and developers
- □ C Development Tools and Libraries
  These tools include core development tools such as automake, gcc and debuggers.
- □ Cloud Management Tools
  Software to managed servers and system images that run in "cloud" environments
- □ Cloud Server Tools
  Software for servers and system images that run in "cloud" environments
- □ Container Management
  Tools for managing Linux containers
- □ D Development Tools and Libraries
  These include development tools and lib such as ldc, geany-tag .
- ☑ Design Suite

Figure 5.11. Software Selection

To configure your software selection, first choose an environment on the left side of the screen. Only one environment can be chosen, even if more are available. Then, on the right side of the screen, select one or more add-ons which you want to install by marking the check boxes next to each add-on.

The list of add-ons is divided into two parts by a horizontal line. Add-ons above this line are defined as part of your chosen environment; if you select a different environment, the add-ons available here will change. The add-ons displayed below the separator are not specific to your chosen environment.

Environments and add-ons are defined using a **`comps.xml`** file in your installation source (for example, in the **`repodata/`** directory on the full Fedora Server installation DVD). Review this file to see exactly which packages will be installed as part of a certain environment or add-on. For more information about the `comps.xml` file, see *Section A.12, “%packages (required) - Package Selection”*.

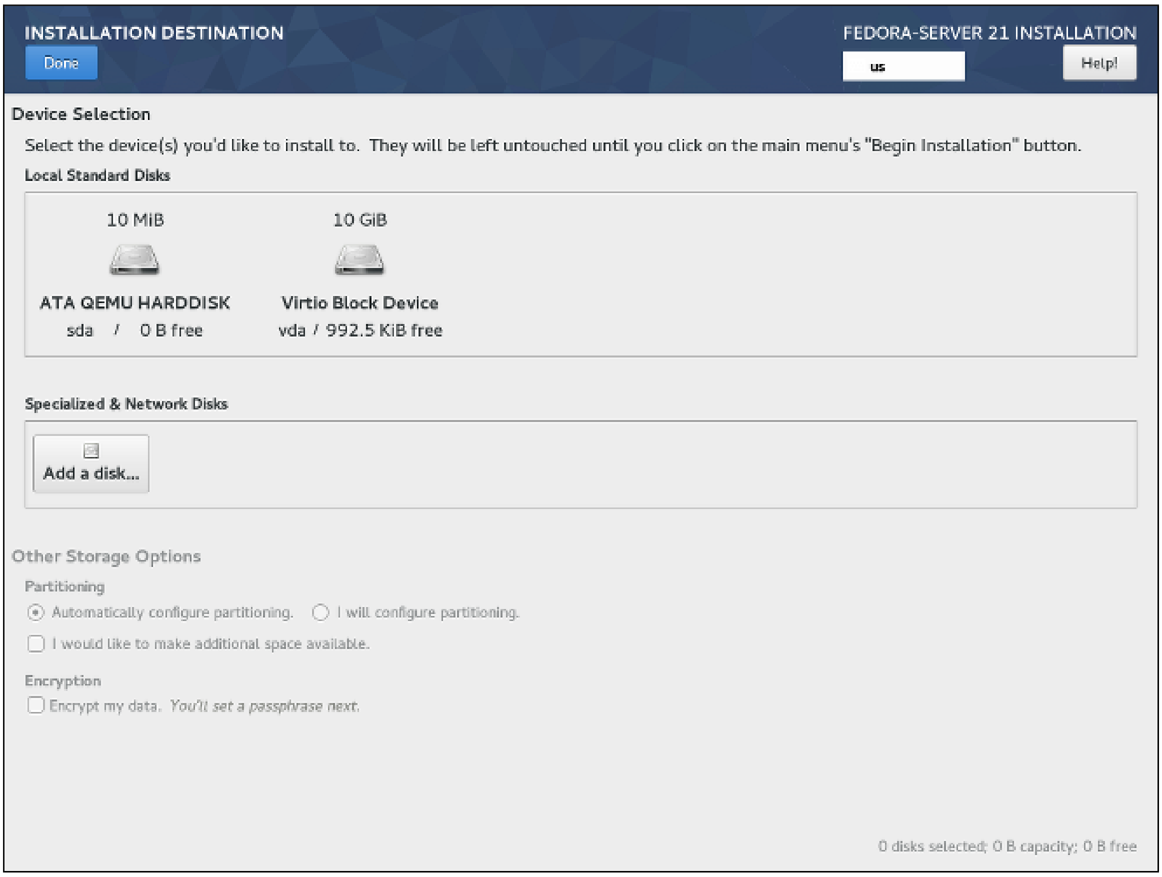
After you finish configuring your software selection, click **Done** in the top left corner to return to *Section 5.4.2, “Installation Summary”*.

5.4.8. Installation Destination

The **Installation Destination** screen allows you to configure storage options - namely, which disks will be used as the installation target for your Fedora installation. At least one disk must always be selected for the installation to proceed.

For information about the theory and concepts behind disk partitioning in Linux, see *Section 5.4.10.6, “Recommended Partitioning Scheme”*.

> **Warning**
>
> If you plan to use a disk which already contains some data - for example, if you want to shrink an existing Microsoft Windows partition and install Fedora as a second system or if you are upgrading a previous release of Fedora, make sure to back up any important data first. Manipulating partitions always carries a risk - if the process is interrupted or fails for any reason (installer error, hardware failure, power outage, etc.), any data already on the disk may become impossible to recover.



Device Selection

Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.

Local Standard Disks

10 MiB	10 GiB
ATA QEMU HARDDISK	Virtio Block Device
sda / 0 B free	vda / 992.5 KiB free

Specialized & Network Disks

Other Storage Options

Partitioning

● Automatically configure partitioning. ○ I will configure partitioning.

□ I would like to make additional space available.

Encryption

□ Encrypt my data. *You'll set a passphrase next.*

0 disks selected; 0 B capacity; 0 B free

Figure 5.12. Installation Destination

In the top part of the screen, all locally available storage devices (SATA, IDE and SCSI hard drives, USB flash drives, etc.) are displayed in the **Local Standard Disks** section. Local disks are detected when the installer starts - any storage devices connected after the installation has started will not be shown.

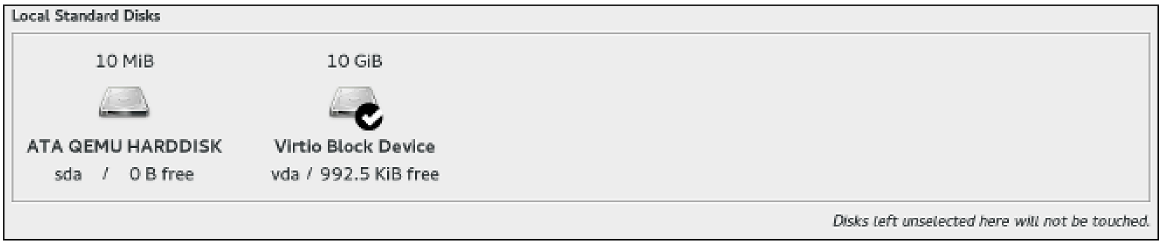
If you need to additional local storage devices, select **I will configure partitioning** and press **Done** to move to *Section 5.4.10, “Manual Partitioning”*. Then, connect any new hard drives you want to make available during the installation, and press the button marked by a circular arrow in the set of controls below the list of mount points on the left side of the screen. In the dialog window that opens, press **Rescan Disks** and wait until the scanning process completes. Then, press **OK** to return to *Section 5.4.8, “Installation Destination”*; all detected disks including any new ones will be displayed in the **Local Standard Disks** section.

The **Specialized & Network Disks** section below shows advanced network storage (such as iSCSI and FCoE disks) currently configured. When you first open this screen, no such devices will be displayed because they can not be automatically detected; to search for network storage devices **Add a disk** button and proceed with *Section 5.4.9, “Installation Destination - Specialized & Network Disks”*. Any network storage you configure will then show up in the **Specialized & Network Disks** the same way local disks are shown above.

All storage devices which will be used to install Fedora have a black circle icon with a white check mark on them. Disks not marked by this icon will not be used during the installation - they will be ignored if you choose automatic partitioning, and they will not be available in manual partitioning.

Important

USB storage devices such as flash drives and external disks will be shown in the **Local Standard Disks** as well, and they will be available for selection the same way internal hard drives are. Make sure to *not* select any removable storage as installation targets unless you really want to do so. If you accidentally use a removable drive to install Fedora and then unplug it, your system will likely become unusable.



Disks left unselected here will not be touched.

Figure 5.13. Unselected and Selected Disk

After you select all disks you want to install Fedora, select one of the two options in the **Other Storage Options** section:

- **Automatically configure partitioning** - If this option is selected, then after you press **Done** in the top left corner of the screen, the installer will determine the total amount of space on all selected disks, and it will create a *Logical Volume Management* (LVM) layout suitable for your system. The specifics of this layout depend on whether your system uses BIOS or UEFI firmware, the total amount of free space on your disks, and the amount of RAM on your system (which determines the size of your swap space).

  With automatic partitioning, you can also select the **I would like to make additional space available** option below. Use this option if you want to reclaim space from an existing partitioning layout - for example, if a disk you want to use already contains a different operating system, and you want to make this system's partitions smaller to allow more room for Fedora. The **Reclaim space** dialog which opens if this option is selected is described later in this section.

- **I will configure partitioning** - Select this option and press **Done** in the top left corner of the screen to configure your system's partitioning layout manually. This requires some knowledge of the theory behind disk partitions and related concepts, but gives you full control over the way the system will be installed. For instructions for manual partitioning, see *Section 5.4.10, "Manual Partitioning"*.

Additionally, you can select **Encrypt my data**; this will encrypt all partitions except the ones needed to boot the system (such as **`/boot`**) using *Linux Unified Key Setup* (LUKS). Encrypting your hard drive is recommended. For detailed information about LUKS encryption, see the *Fedora Security Guide*, available at *http://docs.fedoraproject.org/*.

Warning

> If you lose the LUKS passphrase, any encrypted partitions and the data on them will become completely inaccessible. There is no way to recover a lost passphrase. However, if you perform a Kickstart installation, you can save encryption passphrases and create backup encryption passphrases during the installation. See *Chapter 9, Automating the Installation with Kickstart* for information about Kickstart installations.

To control which one of your selected storage devices will contain the *boot loader*, click the **Full disk summary and bootloader** link in the bottom left corner of the screen, and follow the instructions in *Section 5.4.8.1, "Boot Loader Installation"*. Note that while in most cases it is sufficient to leave the boot loader in the default location, some configurations (for example, systems which require *chain loading* from another boot loader) will require the boot drive to be specified manually.

After you select storage devices, choose between automatic and manual partitioning, configure encryption and boot loader location, press **Done** in the top left corner of the screen. Then, depending on your settings, the following will happen:

- If you chose to encrypt your hard drive, the **Disk Encryption Passphrase** dialog will appear. Enter your chosen passphrase into the **Passphrase** and **Confirm** fields. When you do so, the passphrase will be automatically evaluated and its strength will be displayed, along with suggestions on how to make it stronger if the installer has determined it to be weak. For information about creating strong passwords, see *Fedora Security Guide*.

- If you selected automatic partitioning and the **I would like to make additional space available**, or if there is not enough free space on your selected hard drives to install Fedora, the **Reclaim Space** dialog will appear. This dialog lists all disk devices you have configured and all partitions on those devices. The bottom right corner of the dialog displays information about how much space the system needs for at least a minimal installation and how much space you have reclaimed.

  Warning

  > If you use the **Reclaim Space** dialog to *delete* a partition, all data on that partition will be lost. If you want to preserve your data, use the **Shrink** option, not the **Delete** option.

First, review the displayed list of available storage devices. The **Reclaimable Space** column shows how much space can be reclaimed from each entry. To reclaim space, select a disk or partition, and press either the **Delete** button to delete that partition (or all partitions on a selected disk), or **Shrink** to use free space on a partition while preserving existing data. Alternatively, you can press **Delete all** in the bottom right corner; this will delete all existing partitions on all disks and make this space available to Fedora, but all existing data on all disks will be lost.

After you free enough space for your Fedora installation, press **Reclaim space** to finish.

Note

No changes to any disks will be made until you press **Begin Installation** in *Section 5.4.2, "Installation Summary"*. The **Reclaim Space** dialog only *marks* partitions for resizing or deletion, but no such action is performed immediately.

- If you selected the **I will configure partitioning** option, pressing **Done** will open the **Manual Partitioning** screen. See *Section 5.4.10, "Manual Partitioning"* for further instructions.

5.4.8.1. Boot Loader Installation

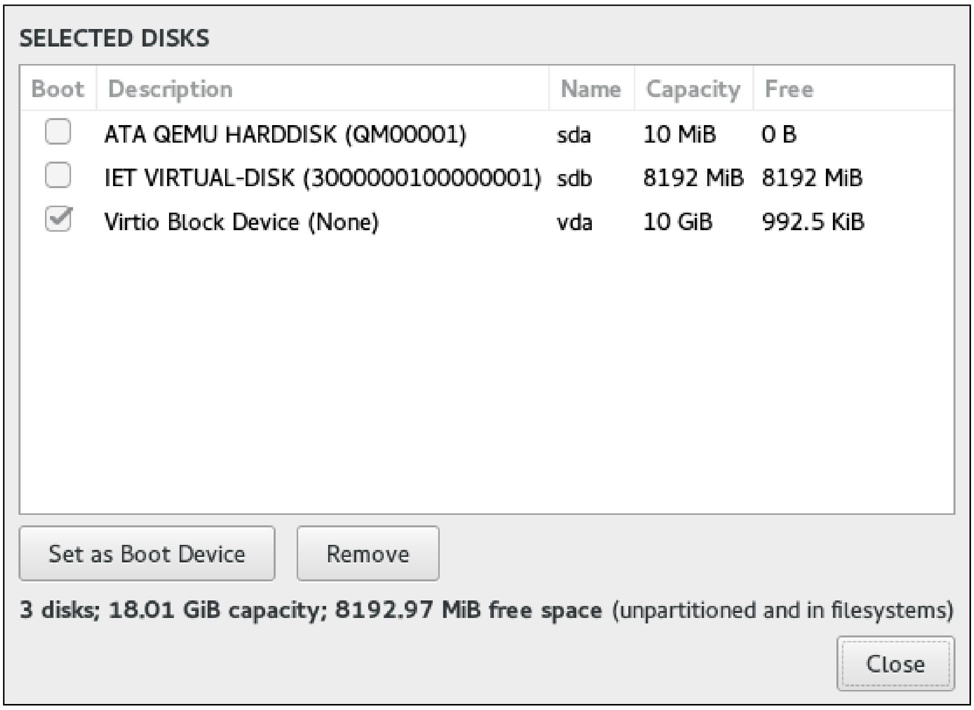
Fedora uses **GRUB2** (GRand Unified Bootloader version 2) as its boot loader. The boot loader is the first program that runs when the computer starts and is responsible for loading and transferring control to an operating system. **GRUB2** can boot any compatible operating system (including Microsoft Windows) and can also use chain loading to transfer control to other boot loaders for unsupported operating systems.

Warning

Installing **GRUB2** may overwrite your existing boot loader.

If you have other operating systems already installed, the Fedora installer will attempt to automatically detect and configure the boot loader to start them. You can manually configure any additional operating systems after you finish the installation, if they are not detected properly. For instructions on editing **GRUB2** configuration, see the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*

If you are installing Fedora system with more than one disk, you may want to manually specify where the bootloader should be installed. Click the **Full disk summary and bootloader** link at the bottom of the **Installation Destination** screen. The **Selected Disks** dialog will appear. The bootloader will be installed on the device of your choice, or on a UEFI system, the `EFI system partition` will be created on that device during guided partitioning.



SELECTED DISKS

Boot	Description	Name	Capacity	Free
□	ATA QEMU HARDDISK (QM00001)	sda	10 MiB	0 B
□	IET VIRTUAL-DISK (3000000100000001)	sdb	8192 MiB	8192 MiB
☑	Virtio Block Device (None)	vda	10 GiB	992.5 KiB

3 disks; 18.01 GiB capacity; 8192.97 MiB free space (unpartitioned and in filesystems)

Figure 5.14. Boot Device Selection

In the **Boot** column, a "tick" icon marks one of the devices as the intended boot device. To change the boot device, select a device from the list and click the **Set as Boot Device** button to install the boot loader there instead. Only one device can be set as the boot device.

To decline installation of a new boot loader, select the device currently marked for boot and click the **Do not install bootloader** button. This will remove the tick and ensure **GRUB2** is not installed on any device.

Warning

If you choose not to install a boot loader for any reason, you will not be able to boot the system directly, and you must use another boot method, such as a stand-alone commercial boot loader application. Use this option only if you are sure you have another way to boot your system.

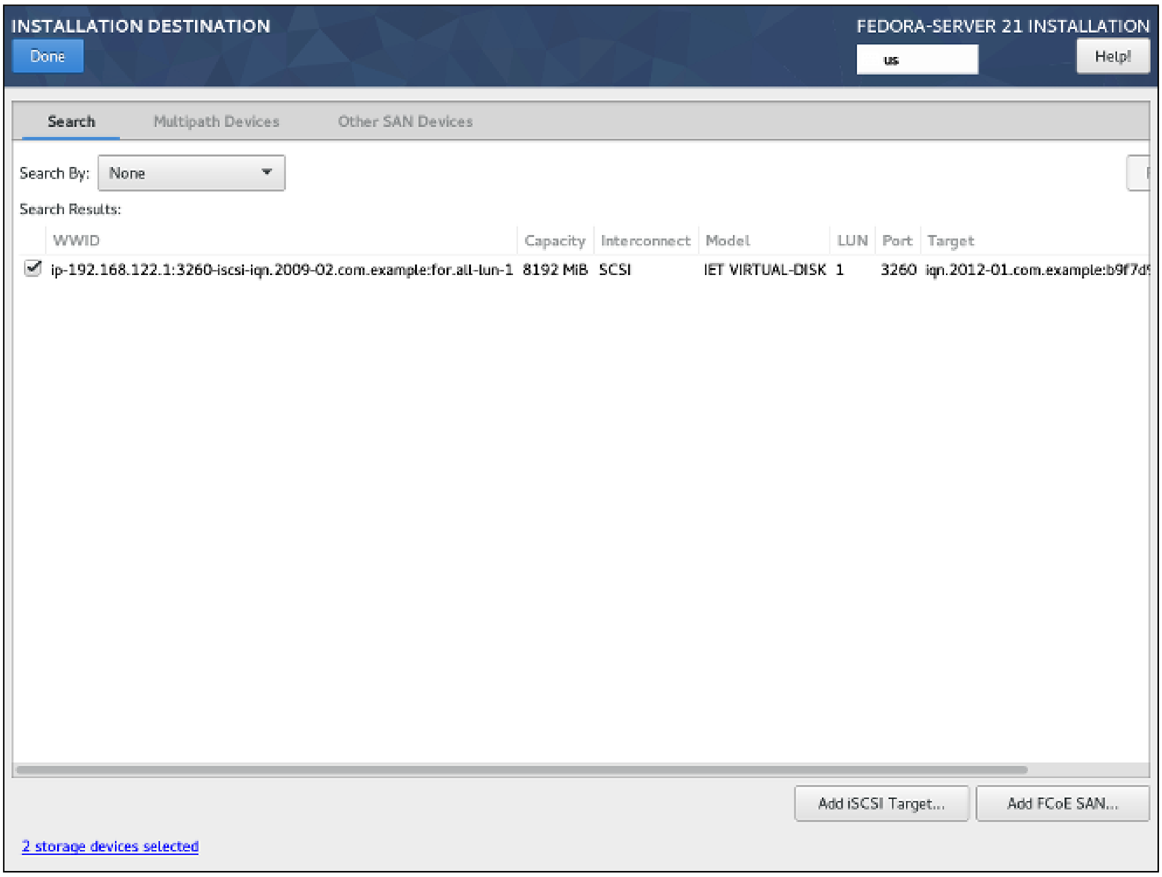
The boot loader may also require a special partition to be created, depending on whether your system uses BIOS or UEFI firmware and also depending on whether the boot drive has a *GUID Partition Table* (GPT) or a **Master Boot Record** (MBR, also known as msdos) label. If you use automatic partitioning, the installer will create this partition if needed. For details, see *Section 5.4.10.6, “Recommended Partitioning Scheme”*.

5.4.9. Installation Destination - Specialized & Network Disks

This part of the **Installation Destination** screen allows you to configure non-local storage devices, namely iSCSI and FCoE storage. This section will mostly be useful to advanced users who have a need for networked disks. For instructions on setting up local hard drives, see *Section 5.4.8, “Installation Destination”*.

Important

This section only explains how to make existing network disks available inside the installer. It does not explain how to set up your network or a storage server, only how to connect to them.



Search By: None

Search Results:

	WWID	Capacity	Interconnect	Model	LUN	Port	Target
☑	ip-192.168.122.1:3260-iscsi-iqn.2009-02.com.example:for.all-lun-1	8192 MiB	SCSI	IET VIRTUAL-DISK	1	3260	iqn.2012-01.com.example:b9f7d

Figure 5.15. Installation Destination - Network Storage Filters

The screen contains a list of all currently available (discovered) network storage devices. When the screen is opened for the first time, the list will be empty in most cases because no network storage has been discovered - the installer makes no attempt at discovering this unless you configure network disks using a Kickstart file.

To add one or more storage devices to the screen so you can search them and use them in the installation, click **Add iSCSI Target** or **Add FCoE SAN** in the bottom right corner of the screen, and follow the instructions in *Section 5.4.9.1, “Add iSCSI Target”* or *Section 5.4.9.2, “Add FCoE SAN”*, depending on which type of network storage you want to add.

Network storage devices successfully discovered and configured by the installer will then be displayed in the main list, along with identifying information such as **Name**, **WWID**, **Model** and **Target**. To sort the list by a specific column (for example **WWID**), click the column's heading.

Note

On lower display resolutions, the list may be too wide to fit on the screen, and some of the columns or buttons may be hidden initially. Use the horizontal scroll bar at the bottom of the list to move your view and see all available table columns and controls.

There are three tabs on the top of the list, which display different information:

Search
: Displays all available devices, regardless of their type, and allows you to filter them either by their *World Wide Identifier* (WWID) or by the port, target, or logical unit number (LUN) at which they are accessed.

Multipath Devices
: Storage devices accessible through more than one path, such as through multiple SCSI controllers or Fiber Channel ports on the same system.

Important

The installation program only detects multipath storage devices with serial numbers that are 16 or 32 characters long.

Other SAN Devices
: Devices available on a Storage Area Network (SAN).

Depending on the tab you are currently in, you can filter the discovered devices by using the **Filter By** field. Some of the filtering options are automatically populated based on discovered devices (for example, if you select **Filter By: Vendor**, another drop-down menu will appear showing all vendors of all discovered devices). Other filters require your input (for example when filtering by WWID), and present you with a text input field instead of a drop-down menu.

In the list (regardless of how it is filtered), each device is presented on a separate row, with a check box to its left. Mark the check box to make the device available during the installation process; this will cause this device (node) to be shown in the **Specialized & Network Disks** section in *Section 5.4.8, “Installation Destination”*. There, you can select the disk as an installation target and proceed with either manual or automatic partitioning.

Note

Devices that you select here are not automatically wiped by the installation process. Selecting a device on this screen does not, in itself, place data stored on the device at risk. Also note that any devices that you do not select here to form part of the installed system can be added to the system after installation by modifying the **`/etc/fstab`** file.

When you have selected the storage devices to make available during installation, click **Done** to return to *Section 5.4.8, “Installation Destination”*.

5.4.9.1. Add iSCSI Target

To use iSCSI storage devices, the installer must be able to discover them as *iSCSI targets* and be able to create an iSCSI session to access them. Both of these steps may require a user name and password for *Challenge Handshake Authentication Protocol* (CHAP) authentication.

You can also configure an iSCSI target to authenticate the iSCSI initiator on the system to which the target is attached (*reverse CHAP*), both for discovery and for the session. Used together, CHAP and reverse CHAP are called *mutual CHAP* or *two-way CHAP*. Mutual CHAP provides the greatest level of security for iSCSI connections, particularly if the user name and password are different for CHAP authentication and reverse CHAP authentication.

Follow the procedure below to add an iSCSI storage target to your system.

Procedure 5.1. Add iSCSI Target

1. Click the **Add iSCSI Target** button in the bottom right corner of the *Section 5.4.9, “Installation Destination - Specialized & Network Disks”* screen. A new dialog window titled **Add iSCSI Storage Target** will open.

2. Enter the IP address of the iSCSI target in the **Target IP Address** field.

3. Provide a name in the **iSCSI Initiator Name** field for the iSCSI initiator in *iSCSI Qualified Name* (IQN) format. A valid IQN entry contains:

   - The string `iqn.` (including the period).

   - A date code specifying the year and month in which your organization's Internet domain or subdomain name was registered, represented as four digits for the year, a dash, and two digits for the month, followed by a period. For example, represent September 2010 as `2010-09.`

   - Your organization's Internet domain or subdomain name, presented in *reverse* order (with the top-level domain first). For example, represent the subdomain storage.example.com as `com.example.storage`.

   - A colon (`:`) followed by a string which uniquely identifies this particular iSCSI initiator within your domain or subdomain. For example, `:diskarrays-sn-a8675309`

   A complete IQN will therefore look as follows:

   ```
   iqn.2010-09.com.example.storage:diskarrays-sn-a8675309
   ```

   An example using the correct format is also displayed below the input field for reference.

   For more information about IQNs, see *3.2.6. iSCSI Names in RFC 3720 - Internet Small Computer Systems Interface (iSCSI)*, available from http://tools.ietf.org/html/rfc3720#section-3.2.6 and *1. iSCSI Names and Addresses in RFC 3721 - Internet Small Computer Systems Interface (iSCSI) Naming and Discovery*, available from http://tools.ietf.org/html/rfc3721#section-1.

4. Specify the type of authentication to use for iSCSI discovery using the **Discovery Authentication Type** drop-down menu. Depending on which type of authentication you selected, additional input fields (such as **CHAP Username** and **CHAP Password** may then become visible. Fill in your authentication credentials; these should be provided by your organization.

5. Click the **Start Discovery** button. The installer will now attempt to discover an iSCSI target based on the information you provided, and if the target requires CHAP or reverse CHAP authentication, it will attempt to use the credentials you provided. This process may take some time (generally less than 30 seconds), depending on your network.

   If the discovery was *not* successful, an error message will be displayed in the dialog window. This message will vary based on which part of the discovery failed. If the installer did not find the target you specified at all, you should check the IP address; if the problem is an authentication error, make sure you entered all CHAP and reverse CHAP credentials correctly and that you have access to the iSCSI target.

   > **Note**
   >
   > The **No nodes discovered** error message may also mean that all nodes on the address you specified are already configured. During discovery, **Anaconda** ignores nodes which have already been added.

   If the discovery was successful, you will see a list of all discovered nodes.

6. Select one or more nodes you want to log in to by marking or unmarking the check box next to each node discovered on the target. Below the list, select again the type of authentication you want to use; you can also select the **Use the credentials from discovery** option if the CHAP/reverse CHAP user name and password you used to discover the target are also valid for logging in to it.

   After selecting all nodes you want to use, click **Log In** to initiate an iSCSI session. **Anaconda** will attempt to log in to all selected nodes. If the login process is succesful, the **Add iSCSI Storage Target** dialog will close, and all nodes you have configured will now be shown in the list of network disks in *Section 5.4.9, “Installation Destination - Specialized & Network Disks”*.

You can repeat this procedure to discover additional iSCSI targets, or to add more nodes from a previously configured target. However, note that once you click the **Start Discovery** button for the first time, you will not be able to change the **iSCSI Initiator Name**. If you made an error when configuring the initiator name, you must restart the installation.

5.4.9.2. Add FCoE SAN

The following procedure explains how to add *Fibre Channel over Ethernet* (FCoE) storage devices and make them available during the installation:

Procedure 5.2. Add FCoE Target

1. Click the **Add FCoE SAN** button in the bottom right corner of *Section 5.4.9, “Installation Destination - Specialized & Network Disks”*. A new dialog window will open.

2. Select the network interface (**NIC**) which is connected to your FCoE switch from the drop-down menu. Note that this network interface must be configured and connected - see *Section 5.4.12, “Network & Hostname”*.

3. Below the **NIC** drop-down menu are two choices:

Use DCB

Data Center Bridging (DCB) is a set of enhancements to the Ethernet protocols designed to increase the efficiency of Ethernet connections in storage networks and clusters. This option should only be enabled for network interfaces that require a host-based DCBX client. Configurations on interfaces that implement a hardware DCBX client should leave this check box empty.

Use auto vlan

This option indicates whether VLAN discovery should be performed. If this box is checked, then the *FCoE Initiation Protocol* (FIP) VLAN discovery protocol will run on the Ethernet interface once the link configuration has been validated. If they are not already configured, network interfaces for any discovered FCoE VLANs will be automatically created and FCoE instances will be created on the VLAN interfaces. This option is enabled by default.

4. After you select which interface and options to use, click **Add FCoE Disk(s)**. Discovered FCoE storage devices will be displayed under the **Other SAN Devices** tab in *Section 5.4.9, “Installation Destination - Specialized & Network Disks”*.

5.4.10. Manual Partitioning

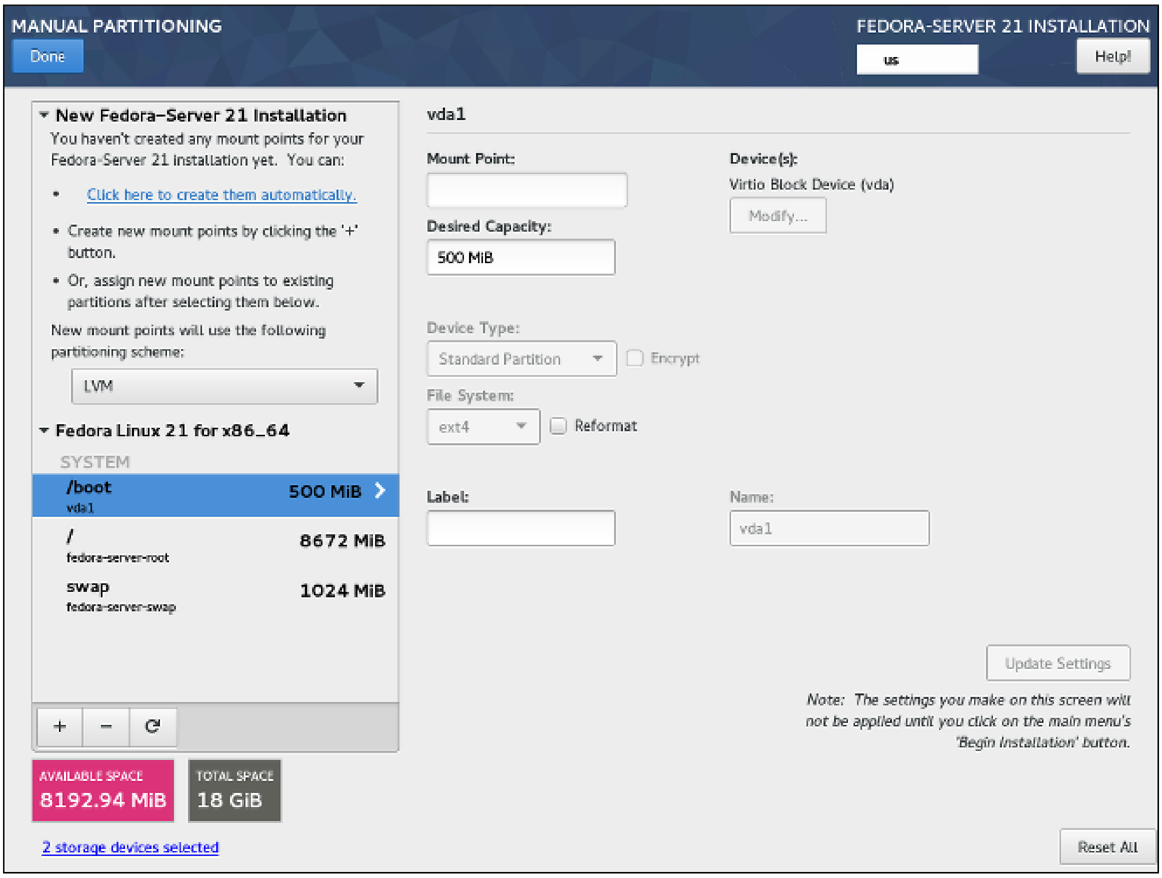
The **Manual Partitioning** screen allows you to create a storage configuration for your Fedora system manually, giving you a greater control over your system's storage.

In most other installers for both Linux and other operating systems, disk partitioning usually takes a "bottom-up" approach. In these installers, you first create underlying devices such as LVM physical volumes, then you create a layout such as LVM on top of them, then you create file systems on top of logical volumes, and the last step is usually assigning a mount point to each volume as needed.

Anaconda uses an opposite approach. First, you create all separate mount points you need, and everything needed to create them (creating a volume group, logical volumes inside it, and physical volumes where the volume group will reside) is performed automatically. You can then adjust the automatic settings as you require.

Note

No permanent changes will be made to your disks during the actual partitioning process. The configuration you have selected will only be written to your system after you press the **Begin installation** button in *Section 5.4.2, “Installation Summary”*.



New Fedora-Server 21 Installation

You haven't created any mount points for your Fedora-Server 21 installation yet. You can:

- Click here to create them automatically.
- Create new mount points by clicking the '+' button.
- Or, assign new mount points to existing partitions after selecting them below.

New mount points will use the following partitioning scheme: LVM

Fedora Linux 21 for x86_64

SYSTEM

/boot vda1	500 MiB
/ fedora-server-root	8672 MiB
swap fedora-server-swap	1024 MiB

AVAILABLE SPACE 8192.94 MiB

TOTAL SPACE 18 GiB

vda1

Mount Point:

Desired Capacity: 500 MiB

Device(s): Virtio Block Device (vda)

Device Type: Standard Partition

File System: ext4

Label:

Name: vda1

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

Figure 5.16. Manual Partitioning

When you first open the **Manual Partitioning** screen, the column on the left side will display all previously existing partitions on all drives which you selected as installation targers in *Section 5.4.8, “Installation Destination”*. If none of the selected drives contain any existing partitions, then a message informing you that no mount points currently exist will appear.

Here, you can choose a partitioning scheme such as **LVM** or **BTRFS** and click the **Click here to create them automatically** to prompt the installer to create a basic partitioning layout; this layout follows the guidelines described in *Section 5.4.10.6, “Recommended Partitioning Scheme”*. The created layout is a basic layout where partition/volume sizes are determined automatically based on the total amount of available space.

Click the **+** button to add a mount point. In the dialog window that opens, choose a mount point such as **/** or **/home**, and the desired capacity for the mount point (such as **10GB** or **500MB**). Note that specifying the mount point is mandatory, but you do not have to specify the capacity at this point; this is useful when adding a mount point which you want to make larger than the current available space permits. Then, click **Add mount point** to add it to the list using the default settings, which means it will be created as a logical volume, and a new volume group will be created for it unless one already exists.

Then, select the newly created mount point in the list on the left side. A set of controls will display on the right side of the screen, allowing you to change its mount point, the device on which it will physically reside, its capacity, file system, etc. When you change any settings, press **Update Settings** on the bottom right. This will save the adjusted configuration; you can now create another mount point, or select a different existing one and adjust its settings as well.

Note

For a description of available device and file system types, see *Section 5.4.10.5, “Device, File System and RAID Types”*.

To remove a mount point, select it in the list and press the **-** button below.

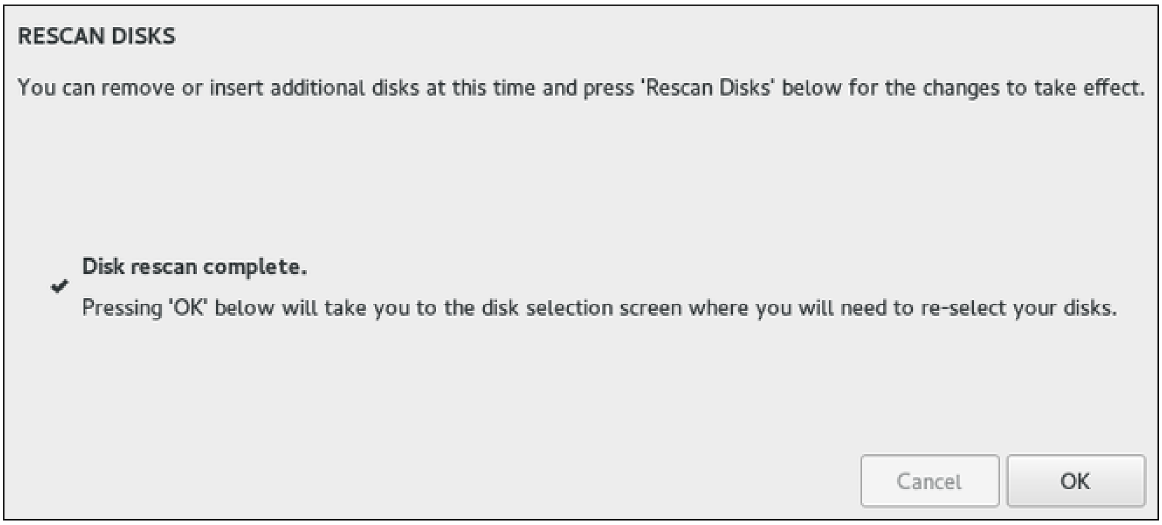
The exact steps for configuring your storage depend on your specific needs and your system configuration. Procedures for creating specific layouts are described further in this chapter. Before you start, you should also review *Section 5.4.10.6, “Recommended Partitioning Scheme”* and *Section 5.4.10.7, “Advice on Partitions”* for a list of requirements and tips for partitioning your disks for Fedora.

Below the list of existing mount points two fields, showing you how much free space is left on your storage devices, and how much total space they have.

Click the ***X* storage devices selected** to view a summary of currently selected storage devices; this may help you with orientation in more complicated storage schemas. Devices displayed here are the ones you have selected in *Section 5.4.8, “Installation Destination”*. If you want to add or remove any storage devices from your configuration, return to that screen and change your selection.

You can press the **Reset All** button in the bottom right corner at any time to reset the storage configuration to the state it was in when you last opened the **Manual Partitioning** screen. This means that if you modify the storage configuration, leave the screen, and then come back, the Reset button will reset the configuration back to the already modified state, discarding only the changes you have made recently, not all changes to the storage configuration since you booted the installer.

To discard all changes, and to also detect any new drives which have not been detected when the installer started (usually when you attached a new drive after you started), press the button marked by a circular arrow in the set of controls below the list of mount points on the left side of the screen. In the dialog window that opens, press **Rescan Disks** and wait until the scanning process completes. Then, press **OK** to return to *Section 5.4.8, “Installation Destination”*; all detected disks including any new ones will be displayed in the **Local Standard Disks** section.


RESCAN DISKS

You can remove or insert additional disks at this time and press 'Rescan Disks' below for the changes to take effect.

✔ Disk rescan complete.

Pressing 'OK' below will take you to the disk selection screen where you will need to re-select your disks.

Figure 5.17. Rescan Disks

After you finish configuring your system storage, press **Done** in the top left corner to save the configuration and return to the **Installation Summary** screen. At this point, the installer will check if your storage configuration is valid. If an error was detected, a message will be displayed at the bottom of the screen. Click the message to open a dialog window explaining what kind of error has been detected (for example, you put **`/boot`** on a Btrfs subvolume, or you did not create a BIOS Boot partition when your system requires one).

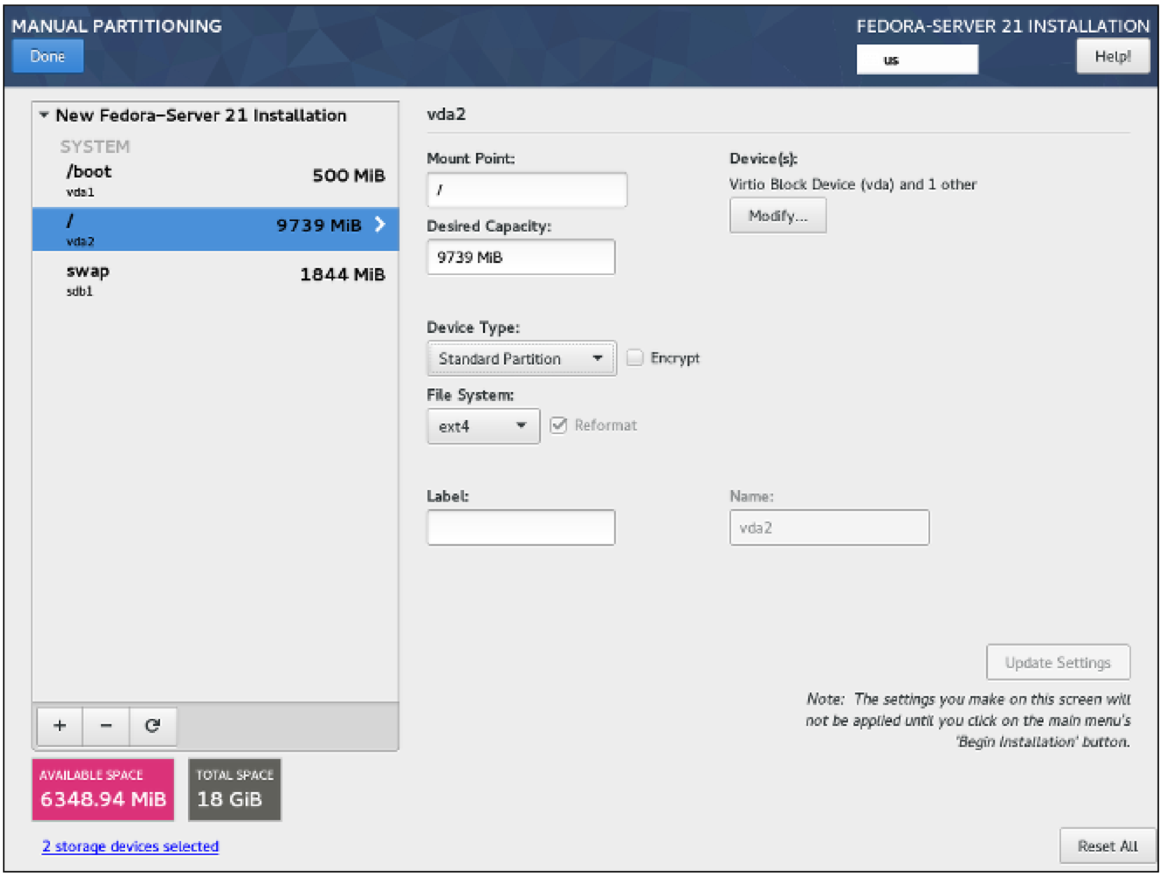
If such a message is displayed, go back and fix any issues found by the installer; otherwise you will not be able to proceed with the installation. You can also press **Done** again to return to the **Installation Summary** anyway, but a storage configuration error will prevent you from starting the actual installation process.

If no error message is displayed and if you made any changes since the last time you have visited this screen, a summary dialog will appear, displaying a detailed list of the changes you made. Review the list and click **Accept Changes** to proceed with *Section 5.4.2, “Installation Summary”*, or click **Cancel & Return to Custom Partitioning** if you want to make any more changes.

5.4.10.1. Creating Standard Partitions

Standard partitions are the most common type of partition, with the widest support across operating systems. For example, Microsoft Windows uses exclusively physical partitions and can not natively work with LVM or Btrfs. Most Fedora partitioning setups will also require at least one standard partition for the **`/boot`** directory, and possibly also another standard partition with the BIOS Boot or EFI System file system to store the boot loader.

See *Appendix B, An Introduction to Disk Partitions* for additional information about the concepts behind physical partitions.



New Fedora-Server 21 Installation

SYSTEM

/boot vda1	500 MiB
/ vda2	9739 MiB
swap sdb1	1844 MiB

vda2

Mount Point: /

Desired Capacity: 9739 MiB

Device(s): Virtio Block Device (vda) and 1 other

Modify...

Device Type: Standard Partition ☐ Encrypt

File System: ext4 ☑ Reformat

Label:

Name: vda2

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

AVAILABLE SPACE 6348.94 MiB

TOTAL SPACE 18 GiB

Figure 5.18. Create Standard Partition

Follow the procedure below to create mount points on standard physical partitions:

Procedure 5.3. Creating Standard Partitions

1. Click the **+** button at the bottom of the list showing existing mount points. A new dialog window will open.

2. In the new dialog window, specify a mount point for which you want to create a separate mount point - for example, `/`. Optionally, specify a size for the partition using standard units such as MB or GB (for example, **`50GB`**). Then, click **Add mount point** to add the mount point and return to the main partitioning screen.

   > **Note**
   >
   > When creating a swap partition, specify the mount point as **`swap`**. For a BIOS Boot partition, use **`biosboot`**. For an EFI System Partition, use **`/boot/efi`**.
   >
   > For information about these partition types, see *Section 5.4.10.6, “Recommended Partitioning Scheme”*.

3. The mount point has now been created using the default settings, which means it has been created as an LVM logical volume. Select the newly created mount point in the left pane to configure it further, and convert it to a physical partition by changing the **Device Type** option to **Standard Partition**. Then, click **Update Settings** in the bottom right corner of the screen.

4. In the **Device(s)** section on the right side of the screen, you can see that the partition has been assigned to one or more hard drives. Click the **Modify** button to configure on which drive this partition will be created.

5. In the **Configure Mount Point** dialog, you can specify which physical devices (disks) this volume *may* reside on. You can select one or more disks which will be used to hold this volume by holding down **`Ctrl`** and clicking each disk in the list. If you select multiple disks here, **Anaconda** will determine where exactly the partition should be created based on how you configured the rest of the installation. If you want to make sure that this partition is placed on a specific hard drive, select only that drive and unselect all others.

   After you finish configuring the partition's location, click **Save** to return to the main **Manual Partitioning** screen.

6. Configure other settings specific to the partition - its **Mount Point**, **Desired Capacity**, and **File System**. Press **Update Settings** to apply any changes to the configuration.

Repeat this procedure for any additional standard partitions you want to create.

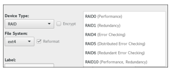
5.4.10.2. Creating Software RAID

Redundant arrays of independent disks (RAIDs) are constructed from multiple storage devices that are arranged to provide increased performance and, in some configurations, greater fault tolerance. See *Section 5.4.10.5, “Device, File System and RAID Types”* a description of different kinds of RAIDs.

A RAID device is created in one step, and disks are added or removed as necessary. One RAID partition per physical disk is allowed for each device, so the number of disks available to the installation program determines which levels of RAID device are available to you. For example, if your system has two hard drives, the installation program will not allow you to create a RAID10 device, which requires 4 separate partitions.

Important

This section only explains how to create software RAID with standard (physical) partitions. However, you can also configure LVM volume groups and Btrfs volumes to use RAID and place their logical volumes or Btrfs subvolumes on top of this RAID array. See *Section 5.4.10.3, “Creating a Logical Volume Managament (LVM) Layout”* and *Section 5.4.10.4, “Creating a Btrfs Layout”* for instructions on creating RAID in LVM and Btrfs.



Device Type: RAID ☐ Encrypt

File System: ext4 ☑ Reformat

Label:

RAID0 (Performance)
RAID1 (Redundancy)
RAID4 (Error Checking)
RAID5 (Distributed Error Checking)
RAID6 (Redundant Error Checking)
RAID10 (Performance, Redundancy)

Figure 5.19. Create Software RAID

Note

RAID configuration options are only visible if you have selected two or more disks for installation. At least two disks are required to create a RAID device, and some RAID layouts will require more. Requirements for different types of RAID are described in *Section 5.4.10.5, “Device, File System and RAID Types”*.

Follow the procedure below to create software RAID:

Procedure 5.4. Creating Software RAID

1. Click the **+** button at the bottom of the list showing existing mount points. A new dialog window will open.

2. In the new dialog window, specify a mount point for which you want to create a separate software RAID partition - for example, `/`. Optionally, specify a size for the new partition using standard units such as MB or GB (for example, **`50GB`**). Then, click **Add mount point** to add the mount point and return to the main partitioning screen.

   Note

   When creating a mount point for swap on software RAID, specify the mount point as **`swap`**.

3. The mount point has now been created using the default settings, which means it has been created as an LVM logical volume. Select the newly created mount point in the left pane to configure it further, and convert it to a software RAID partition by changing the **Device Type** option to **RAID**.

4. Choose a RAID type from the **RAID Level** drop-down menu. Available RAID types and their requirements are described in *Section 5.4.10.5, “Device, File System and RAID Types”*.

5. In the **Device(s)** section on the right side of the screen, you can see that the partition has been assigned to several physical disks. Click the **Modify** button to configure on which drives this partition will be created.

6. In the **Configure Mount Point** dialog, you can specify which physical devices (disks) this partition *may* reside on. You can select one or more disks which will be used to hold this partition by holding down **`Ctrl`** and clicking each disk in the list. If you want to make sure that this partition is placed on a specific set of hard drives, select only those drives and unselect all others.

   After you finish configuring the partition's location, click **Save** to return to the main **Manual Partitioning** screen.

7. Configure other settings specific to the partition - its **Mount Point**, **Desired Capacity**, and **File System**. Press **Update Settings** to apply any changes to the configuration.

Repeat this procedure for any additional standard partitions with software RAID you want to create.

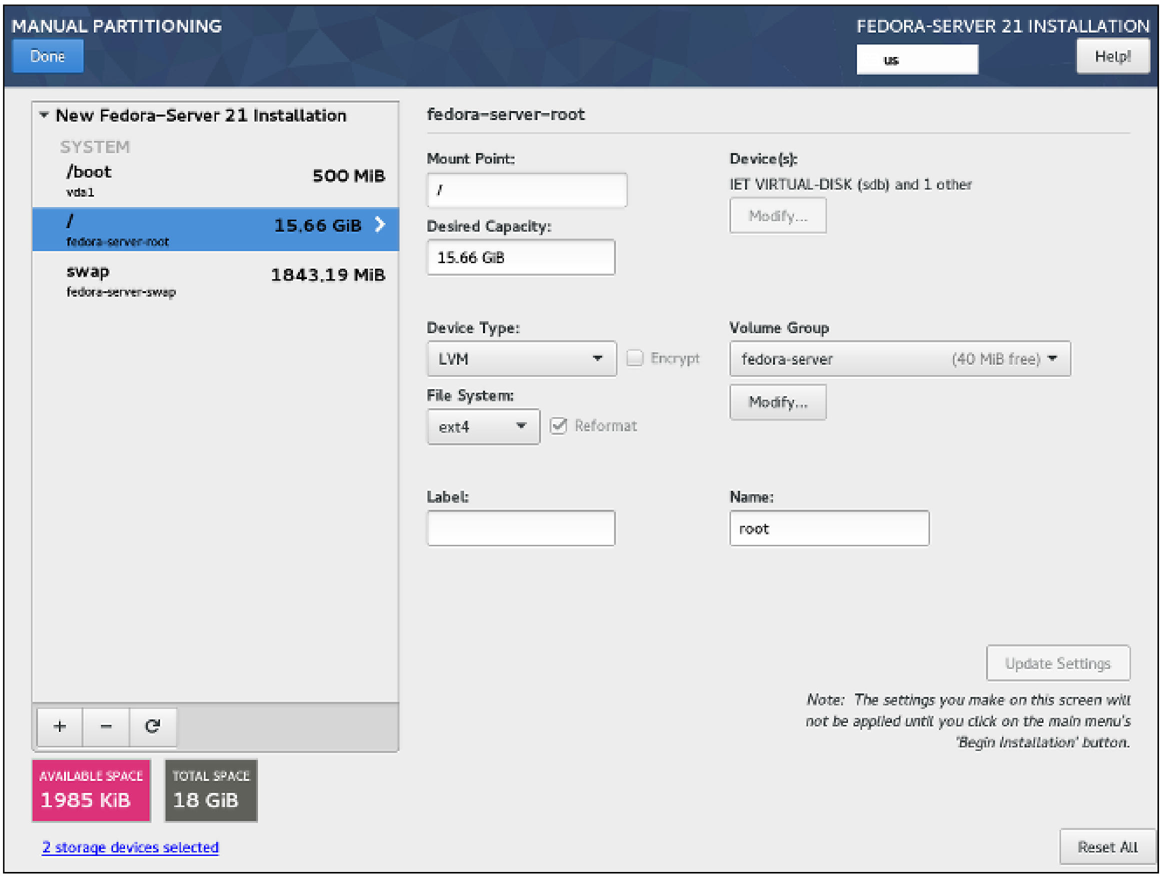
5.4.10.3. Creating a Logical Volume Managament (LVM) Layout

Logical Volume Management (LVM) presents a simple logical view of underlying physical storage space, such as hard drives or LUNs. Partitions on physical storage are represented as *physical volumes* that can be grouped together into *volume groups*. Each volume group can be divided into multiple *logical volumes*, each of which is analogous to a standard disk partition. Therefore, LVM logical volumes function as partitions which can span multiple physical disks.

See *Appendix C, Understanding LVM* for additional information about the concepts behind Logical Volume Management.

Warning

Some partition types - notably the **`/boot`** directory and the BIOS Boot and EFI partitions - can not be placed on logical volumes. Use standard physical volumes for them. See *Section 5.4.10.6, “Recommended Partitioning Scheme”* for more information.



New Fedora-Server 21 Installation

SYSTEM

/boot vda1	500 MiB
/ fedora-server-root	15.66 GiB
swap fedora-server-swap	1843.19 MiB

AVAILABLE SPACE 1985 KiB

TOTAL SPACE 18 GiB

fedora-server-root

Mount Point: /

Desired Capacity: 15.66 GiB

Device(s): IET VIRTUAL-DISK (sdb) and 1 other

Device Type: LVM ☐ Encrypt

File System: ext4 ☑ Reformat

Volume Group: fedora-server (40 MiB free)

Label:

Name: root

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

Figure 5.20. Create LVM Logical Volume

Follow the procedure below to create LVM logical volumes and volume groups.

Procedure 5.5. Creating LVM Logical Volumes and Groups

1. Click the **+** button at the bottom of the list showing existing mount points. A new dialog window will open.

2. In the new dialog window, specify a mount point for which you want to create a separate logical volume - for example, `/`. Optionally, specify a size for the volume using standard units such as MB or GB (for example, **`50GB`**). Then, click **Add mount point** to add the volume and return to the main partitioning screen.

   ### Note

   When creating a mount point for swap on LVM, specify the mount point as **swap**.

3. The mount point has now been created using the default settings, which means it has been created as an LVM logical volume, and a volume group has been created to contain it. Select the newly created mount point in the left pane to configure it further. If you want to use thin provisioning for this volume, change the **Device Type** option to **LVM Thin Provisioning**.

4. In the **Volume Group** menu, you can see that the volume has been assigned to an automatically created volume group, which is named after the Fedora variant you are installing (for example,

`fedora-server`. Click the **Modify** button under the drop-down menu to access the volume group settings.

5. In the **Configure Volume Group** dialog, you can change the volume group's name, its **RAID level** (see *Section 5.4.10.5, “Device, File System and RAID Types”* for information about available RAID types), and you can also specify which physical devices (disks) this volume group should reside on. You can select one or more disks which will be used to hold this volume group by holding down **`Ctrl`** and clicking each disk in the list.

   Note

   If you select a redundant RAID type (such as **RAID1 (Redundancy)**), the volume group will take up twice its actual size on your disks. A 5 GB volume group with RAID1 will take up 10 GB of space.

   You can also make sure that the volume group is encrypted by selecting the **Encrypt** option; this will enable LUKS encryption for the entire volume group. See the *Fedora Security Guide*, available at *http://docs.fedoraproject.org/*, for information about LUKS disk encryption.

   Additionally, you can set a fixed size for the volume group by selecting the **Fixed** option from the **Size policy** menu and entering a size for the volume group.

   After you finish configuring the volume group settings, click **Save** to return to the main **Manual Partitioning** screen.

   Note

   The configuration dialog does not allow you to specify the size of the volume group's *physical extents*. The size will always be set to the default value of 4 MiB. If you want to create a volume group with different physical extents, create it manually by switching to an interactive shell and using the **`vgcreate`** command, or use a Kickstart file with the **`volgroup --pesize=`*`size`*** command.

6. If you need to create more than one volume group, open the **Volume Group** drop-down menu and select the **Create a new volume group** option. A new dialog window will open, identical to the one described in the previous step. Again, select a name, storage devices, encryption settings, RAID level and size policy for the new group, and click **Save**. The new volume group will then become available in the **Volume Group** drop-down menu; you can then go through your existing mount points and change this setting to assign them to a different volume group.

7. Configure other settings specific to the logical volume - its **Mount Point**, **Desired Capacity**, **File System**, and **Name**. Press **Update Settings** to apply any changes to the configuration.

Repeat this procedure for any additional logical volumes you want to create. Note that when creating additional LVM logical volumes, a new volume group is not automatically created each time; instead, any additional volumes are assigned to an existing group.

For each mount point you create, review its settings and make sure that it is assigned to the correct group, that it has sufficient capacity, and that it has a descriptive name so you can identify the volume later if you need to.

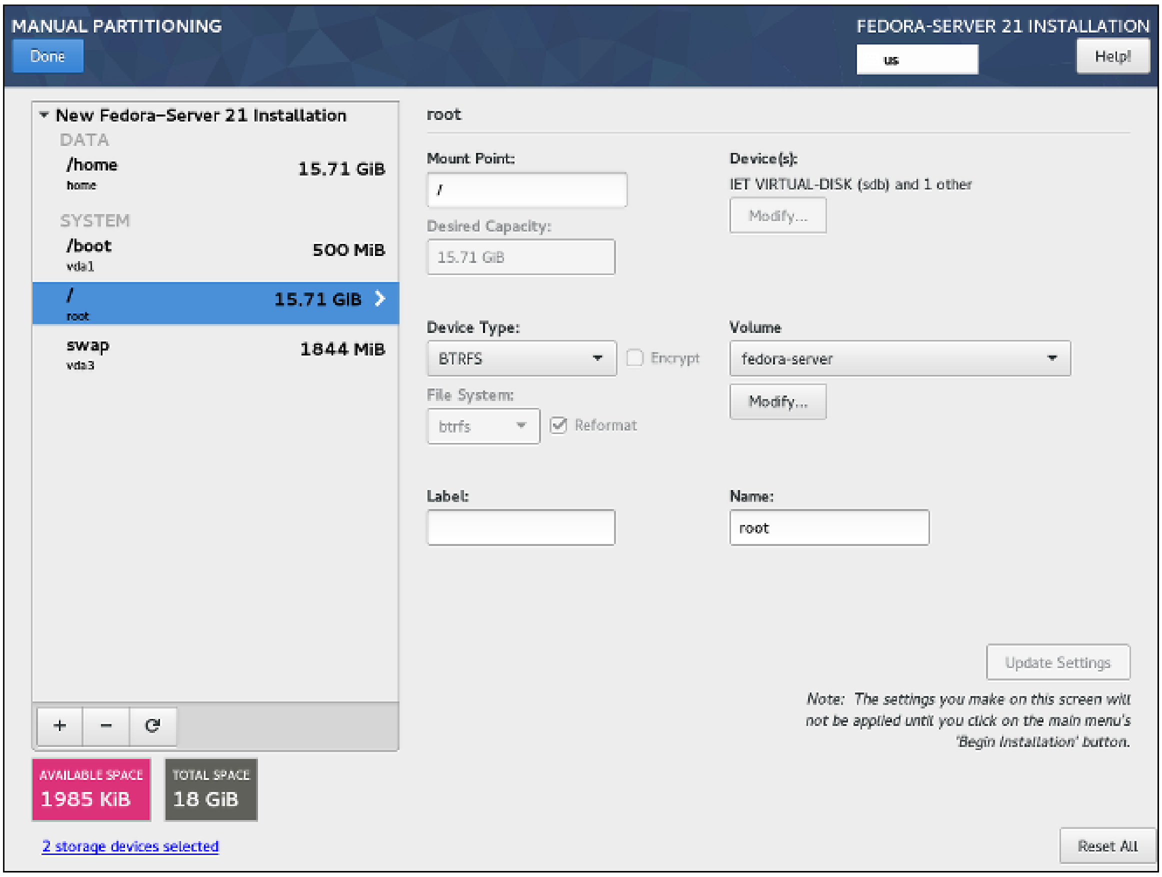
5.4.10.4. Creating a Btrfs Layout

Btrfs is a type of file system, but it has several features characteristic of a storage device. It is designed to make the file system tolerant of errors, and to facilitate the detection and repair of errors when they occur. It uses checksums to ensure the validity of data and metadata, and maintains snapshots of the file system that can be used for backup or repair.

Creating a Btrfs layout is somewhat similar to LVM (described in *Section 5.4.10.3, “Creating a Logical Volume Managament (LVM) Layout”*) with slightly different terminology. A Btrfs *volume* is the equivalent of an LVM volume group, and a Btrfs *subvolume* is similar to a LVM logical volume. An important difference to note is how **Anaconda** reports sizes for separate mount points: For LVM, the exact size of each logical volume is shown next to each mount point in the left pane, while with Btrfs, the total size of the entire volume is shown next to each subvolume.

Warning

Some partition types - notably the `/boot` and `/usr` directories and the BIOS Boot and EFI partitions - can not be placed on Btrfs subvolumes. Use standard physical volumes for them (or an LVM logical volume for `/usr`). See *Section 5.4.10.6, “Recommended Partitioning Scheme”* for more information.



▾ New Fedora-Server 21 Installation

DATA

/home home	15.71 GiB

SYSTEM

/boot vda1	500 MiB
/ root	15.71 GiB ›
swap vda3	1844 MiB

AVAILABLE SPACE 1985 KiB

TOTAL SPACE 18 GiB

2 storage devices selected

root

Mount Point: /

Desired Capacity: 15.71 GiB

Device(s): IET VIRTUAL-DISK (sdb) and 1 other

Device Type: BTRFS ▾ □ Encrypt

Volume: fedora-server ▾

File System: btrfs ▾ ☑ Reformat

Label:

Name: root

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

Figure 5.21. Create Btrfs Subvolume

Follow the procedure below to create Btrfs volumes and subvolumes:

Procedure 5.6. Creating Btrfs Subvolumes and Volumes

1. Click the **+** button at the bottom of the list showing existing mount points. A new dialog window will open.

2. In the new dialog window, specify a mount point for which you want to create a separate logical volume - for example, `/`. Optionally, specify a size for the volume using standard units such as MB or GB (for example, **`50GB`**). Then, click **Add mount point** to add the volume and return to the main partitioning screen.

   > **Note**
   >
   > When creating a mount point for swap on Btrfs, specify the mount point as **`swap`**.

3. The mount point has now been created using the default settings, which means it has been created as an LVM logical volume. Select the newly created mount point in the left pane to configure it further, and convert it to a Btrfs subvolume by changing the **Device Type** option to **Btrfs**. Then, click **Update Settings** in the bottom right corner of the screen.

4. In the **Volume** menu, you can see that the subvolume has been assigned to an automatically created volume, which is named after the Fedora variant you are installing (for example, **`fedora-server00`**. Click the **Modify** button under the drop-down menu to access the volume settings.

5. In the **Configure Volume** dialog, you can change the volume's name, its **RAID level** (see *Section 5.4.10.5, “Device, File System and RAID Types”* for information about available RAID types), and you can also specify which physical devices (disks) this volume should reside on. You can select one or more disks which will be used to hold this volume by holding down **`Ctrl`** and clicking each disk in the list.

   > **Note**
   >
   > If you select a redundant RAID type (such as **RAID1 (Redundancy)**), the volume will take up twice its actual size on your disks. A 5 GB volume with RAID1 will take up 10 GB of space.

   You can also make sure that the volume is encrypted by selecting the **Encrypt** option; this will enable LUKS encryption for the entire volume. See the *Fedora Security Guide*, available at *http://docs.fedoraproject.org/*, for information about LUKS disk encryption.

   Additionally, you can set a fixed size for the volume by selecting the **Fixed** option from the **Size policy** menu and entering a size for the volume group.

   After you finish configuring the Btrfs volume settings, click **Save** to return to the main **Manual Partitioning** screen.

6. If you need to create more than one Btrfs volume, open the **Volume** drop-down menu and select the **Create a new volume group** option. A new dialog window will open, identical to the one described in the previous step. Again, select a name, storage devices, encryption settings, RAID

level and size policy for the new volume, and click **Save**. The new volume will then become available in the **Volume Group** drop-down menu; you can then go through your existing mount points and change this setting to assign them to a different volume.

7. Configure other settings specific to the subvolume - its **Mount Point**, **Desired Capacity**, **File System**, and **Name**. Press **Update Settings** to apply any changes to the configuration.

Repeat this procedure for any additional Btrfs subvolumes you want to create. Note that when creating additional subvolumes, a new volume is not automatically created each time; instead, any additional subvolumes are assigned to an existing volume.

For each mount point you create, review its settings and make sure that it is assigned to the correct volume, that it has sufficient capacity, and that it has a descriptive name so you can identify the subvolume later if you need to.

5.4.10.5. Device, File System and RAID Types

Fedora supports multiple types of devices and file systems. The lists below offer a short description of each available device, file system and RAID type and notes on their usage.

To select a device type or a file system of a partition or a logical volume, select it in the list in *Section 5.4.10, “Manual Partitioning”* and select a **Device Type** and a **File System** from their respective drop-down menus on the right side of the screen. Then, click **Update Settings** and repeat this process for all mount points you want to modify.

To configure software RAID, make sure that you have enough physical hard drives selected as installation targets (the number of separate drives required for each type of RAID is noted in its description). Then, choose a RAID level when creating or modifying a Btrfs volume or LVM volume group, or select **Software RAID** as the device type to create software RAID with standard partitions. For detailed instructions, see *Section 5.4.10.4, “Creating a Btrfs Layout”*, *Section 5.4.10.3, “Creating a Logical Volume Managament (LVM) Layout”*, and *Section 5.4.10.2, “Creating Software RAID”* as needed.

Device Types

- **Standard Partition** - A standard partition can contain a file system or swap space. Standard partitions are most commonly used for **`/boot`** and the BIOS Boot and EFI System partitions. LVM logical volumes or Btrfs subvolumes are recommended for most other uses. See *Appendix B, An Introduction to Disk Partitions* for additional information about the concepts behind physical partitions.

- **LVM** - Choosing **LVM** as the **Device Type** creates an LVM logical volume and a volume group to contain it (unless one already exists, in which case the new volume is assigned to the existing group). LVM can improve performance when using physical disks and allows you to use multiple disks for a single mount point. For information on how to create a logical volume, see *Section 5.4.10.3, “Creating a Logical Volume Managament (LVM) Layout”*. Also see *Appendix C, Understanding LVM* for some additional information about LVM in general.

- **LVM Thin Provisioning** - Using thin provisioning, you can manage a storage pool of free space, known as a *thin pool*, which can be allocated to an arbitrary number of devices when needed by applications. The thin pool can be expanded dynamically when needed for cost-effective allocation of storage space.

- **RAID** - Creating two or more software RAID partitions allows you to create a software RAID device. One RAID partition is assigned to each disk on the system. See *Section 5.4.10.2, “Creating Software RAID”* for instructions on creating software RAID.

- **BTRFS** - Btrfs is a file system with several device-like features. It is capable of addressing and managing more files, larger files, and larger volumes than the ext2, ext3, and ext4 file systems. See *Section 5.4.10.4, “Creating a Btrfs Layout”* for more information about creating Btrfs volumes.

File Systems

- **ext4** - The ext4 file system is based on the ext3 file system and features a number of improvements. These include support for larger file systems and larger files, faster and more efficient allocation of disk space, no limit on the number of subdirectories within a directory, faster file system checking, and more robust journaling. Ext4 is the default and recommended file system used by Fedora Workstation and Cloud. The maximum supported size of a single ext4 file system is 50 TB.

- **ext3** - The ext3 file system is based on the ext2 file system and has one main advantage - journaling. Using a journaling file system reduces time spent recovering a file system after a crash, as there is no need to check the file system for metadata consistency by running the **`fsck`** utility every time a crash occurs.

- **ext2** - An ext2 file system supports standard Unix file types, including regular files, directories, or symbolic links. It provides the ability to assign long file names, up to 255 characters.

- **swap** - Swap partitions are used to support virtual memory. In other words, data is written to a swap partition when there is not enough RAM to store the data your system is processing. A swap partition should always be created; see *Section 5.4.10.6, “Recommended Partitioning Scheme”* for details such as the recommended size.

- **xfs** - XFS is a highly scalable, high-performance file system that supports file systems up to 16 exabytes (approximately 16 million terabytes), files up to 8 exabytes (approximately 8 million terabytes), and directory structures containing tens of millions of entries. XFS also supports metadata journaling, which facilitates quicker crash recovery. The maximum supported size of a single XFS file system is 500 TB. Starting with Fedora 22, XFS is the default and recommended file system on Fedora Server.

Important

Note that the size of an XFS file system can not currently be reduced without destroying and recreating the file system. If you expect that you will need to adjust the sizes of your file systems often, using XFS is not recommended, as it makes administration substantially more time-consuming.

- **vfat** - The VFAT file system is a Linux file system that is compatible with Microsoft Windows long file names on the FAT file system.

- **BIOS Boot** - A very small partition required for booting from a device with a GUID partition table (GPT) on BIOS systems and UEFI systems in BIOS compatibility mode. See *Section 5.4.10.6, “Recommended Partitioning Scheme”* for details.

- **EFI System Partition** - A small partition required for booting a device with a GUID partition table (GPT) on a UEFI system. See *Section 5.4.10.6, “Recommended Partitioning Scheme”* for details.

Software RAID Types

- **RAID0 (Performance)** - Distributes data across multiple disks. Level 0 RAID offers increased performance over standard partitions and can be used to pool the storage of multiple disks into one large virtual device. Note that Level 0 RAIDs offer no redundancy and that the failure of one device in the array destroys data in the entire array. RAID 0 requires at least two disks.

- **RAID1 (Redundancy)** - Mirrors all data from one partition onto one or more other disks. Additional devices in the array provide increasing levels of redundancy. RAID 1 requires at least two disks.

- **RAID4 (Error Checking)** - Distributes data across multiple disks and uses one disk in the array to store parity information which safeguards the array in case any disk within the array fails. Because all parity information is stored on one disk, access to this disk creates a "bottleneck" in the array's performance. Level 4 RAID requires at least three disks.

- **RAID5 (Distributed Error Checking)** - Distributes data and parity information across multiple disks. Level 5 RAIDs therefore offer the performance advantages of distributing data across multiple disks, but do not share the performance bottleneck of level 4 RAIDs because the parity information is also distributed through the array. RAID 5 requires at least three disks.

- **RAID6 (Redundant Error Checking)** - Level 6 RAIDs are similar to level 5 RAIDs, but instead of storing only one set of parity data, they store two sets. RAID 6 requires at least four disks.

- **RAID10 (Performance, Redundancy)** - Level 10 RAIDs are nested RAIDs or hybrid RAIDs. They are constructed by distributing data over mirrored sets of disks. For example, a level 10 RAID array constructed from four RAID partitions consists of two mirrored pairs of striped partitions. RAID 10 requires at least four disks.

5.4.10.6. Recommended Partitioning Scheme

In most cases, at least the following mount points should always be created:

`/boot` - 500 MB

This partition contains the operating system kernel, which allows Fedora to boot. It also contains other files used during the bootstrap process. Due to the limitations of most firmware, creating a separate, small standard partition for this directory is recommended. In most scenarios, a 500 MB `/boot` partition is adequate.

> **Important**
>
> If your system has a hardware RAID controller, be aware that some BIOS types do not support booting from it. In that case, the `/boot` partition must be created on a partition outside of the RAID array, such as on a separate hard drive.
>
> Also note that the `/boot` directory can not be placed on a LVM logical volume or a Btrfs subvolume. Use a standard partition.

`/` (root) - 10 GB

This is where the root directory is located. The root directory is the top level of the directory structure. By default, all files are written to this partition unless a different partition is mounted in the path being written to (for example, `/boot` or `/home`). If you follow the recommended scheme described in this section, this will be the partition where most software packages will be installed.

For a minimal installation, a 5 GB root partition will be sufficient. However, for most common installations which include extra packages and a graphical user interface, the root partition should be at least 10 GB; with 20 GB being sufficient for most common use cases.

Note

The `/` mount point is the top of the Linux Filesystem Hierarchy, and is referred to as the *root file system*, or root. The **`/root`** directory, sometimes pronounced "*slash-root*", is the home directory for the `root` user.

`/home` - at least 10 GB

To store user data separately from system data, create a dedicated mount point for the **`/home`** directory. This partition should be sized based on the amount of data that will be stored locally, number of users, and so on. This will allow you to upgrade or reinstall Fedora without erasing user data files. During the installation, a separate **`/home`** partition will be created if there are 50 GB or more free space for your Fedora installation.

When using Fedora as a workstation for normal use with a graphical environment, this mount point should have the most disk space assigned to it, as it will likely hold the most data (user settings, images, videos, etc).

swap - based on your system parameters

Swap partitions support virtual memory; data is written to them when there is not enough RAM to store the data your system is processing. This partition's size is a function of system memory workload, not total system memory, and therefore is not equal to the total system memory size. Therefore, it is important to analyze what applications a system will be running and the load those applications will serve in order to determine the system memory workload. Application providers and developers should be able to provide some guidance.

When the system runs out of swap space, the kernel terminates processes as the system RAM memory is exhausted. Configuring too much swap space results in storage devices being allocated but idle and is a poor use of resources. Too much swap space can also hide memory leaks. The maximum size for a swap partition and other additional information can be found in the `mkswap(8)` man page.

The table below provides the recommended size of a swap partition depending on the amount of RAM in your system and whether you want sufficient memory for your system to hibernate. If you let the installation program partition your system automatically, the swap partition size will be established using these guidelines. Automatic partitioning setup assumes hibernation is not in use, and the maximum size of the swap partition is limited to 10% of the total size of the hard drive. If you want to set up enough swap space to allow for hibernation, or if you want to set the swap partition size to more than 10% of the system's storage space, you must edit the partitioning layout manually.

Table 5.2. Recommended System Swap Space

Amount of RAM in the system	Recommended swap space	Recommended swap space if allowing for hibernation
less than 2 GB	2 times the amount of RAM	3 times the amount of RAM
2 GB - 8 GB	Equal to the amount of RAM	2 times the amount of RAM
8 GB - 64 GB	0.5 times the amount of RAM	1.5 times the amount of RAM
more than 64 GB	workload dependent	hibernation not recommended

At the border between each range listed above (for example, a system with 2 GB, 8 GB, or 64 GB of system RAM), discretion can be exercised with regard to chosen swap space and hibernation support. If your system resources allow for it, increasing the swap space may lead to better performance.

Distributing swap space over multiple storage devices - particularly on systems with fast drives, controllers and interfaces - also improves swap space performance.

BIOS Boot (1 MB) or EFI System Partition (200 MB)

The **GRUB2** boot loader can be installed either in the *Master Boot Record* (MBR) or the *GUID Partition Table* (GPT) of the boot device. In order to determine which of these methods to use, the installation program considers the following variations:

Systems with BIOS firmware and UEFI systems in BIOS compatibility mode

If the disk is already formatted, the partitioning scheme is retained. If the disk is not formatted, or you have erased all existing partitions from the disk, the installer will choose the following:

- MBR if the size of the disk is less than 2 TB (terabytes)
- GPT if the size of the disk is more than 2 TB

Note

You can force the installer to use GPT on disks smaller than 2 TB by using the **`inst.gpt`** boot option as described in *Chapter 8, Boot Options*. However, the opposite is not possible - you can not use MBR on disks larger than 2 TB.

You need to create a *BIOS Boot* partition with a size of 1 MB to install on a system with BIOS firmware if the disk containing the boot loader uses GPT. If the disk uses a MBR, no special partition is necessary on a BIOS system.

Systems with UEFI firmware

Only GPT is allowed on UEFI systems. In order to install on a formatted disk with a MBR, it must be reformated and relabeled. All data currently on the disk will be lost.

UEFI-based systems require an *EFI System Partition* at least 50 MB in size (recommended size is 200 MB), regardless of the partitioning scheme.

Important

If your system requires either a BIOS Boot partition or an EFI System Partition based on the requirements detailed above, this partition must be created as a standard physical partition. It can not reside on an LVM volume or a Btrfs subvolume.

Also note that if your system does not require any of these partitions, they will not be shown in the **File System** menu in mount point options.

Many systems have more partitions than the minimum listed above. Choose partitions based on your particular needs. See *Section 5.4.10.7, “Advice on Partitions”* for additional information and advice.

Note

Only assign storage capacity to those partitions you require immediately. You may allocate free space at any time, to meet needs as they occur.

If you are not sure how best to configure the partitions for your computer, accept the automatic default partition layout provided by the installation program as described in *Section 5.4.8, “Installation Destination”*.

5.4.10.7. Advice on Partitions

There is no best way to partition every system; the optimal setup depends on how you plan to use the system being installed. However, the following tips may help you find the optimal layout for your needs:

- Consider encrypting any partitions and volumes which might contain sensitive data. Encryption prevents unauthorized people from accessing the data on the partitions, even if they have access to the physical storage device. In most cases, you should at least encrypt the **`/home`** partition, which contains user data.

- In some cases, creating separate mount points for directories other than `/`, **`/boot`** and **`/home`** may be useful; for example, on a server running a **MySQL** database, having a separate mount point for **`/var/lib/mysql`** will allow you to preserve the database during a reinstallation without having to restore it from backup afterwards. However, having unnecessary separate mount points will make storage administration more difficult.

- Some special restrictions apply to certain directories with regards on which partitioning layouts can they be placed. Notably, the **`/boot`** directory must always be on a physical partition (not on an LVM volume or a Btrfs subvolume), and **`/usr`** can not be on a Btrfs subvolume.

- If you are new to Linux, consider reviewing the *Linux Filesystem Hierarchy Standard* at *http://refspecs.linuxfoundation.org/FHS_2.3/fhs-2.3.html* for information about various system directories and their contents.

- Each kernel installed on your system requires approximately 20 MB on the **`/boot`** partition. The default partition size of 500 MB for **`/boot`** should suffice for most common uses; increase the size of this partition if you plan to keep many kernels installed at the same time.

- The **`/var`** directory holds content for a number of applications, including the **Apache** web server, and is used by the **DNF** package manager to temporarily store downloaded package updates. Make sure that the partition or volume containing **`/var`** has at least 3 GB.

- The contents of the **`/var`** directory usually change very often. This may cause problems with older solid state drives (SSDs), as they can handle a lower number of read/write cycles before becoming unusable. If your system root is on an SSD, consider creating a separate mount point for **`/var`** on a classic (platter) HDD.

- The **`/usr`** directory holds the majority of software on a typical Fedora installation. The partition or volume containing this directory should therefore be at least 5 GB for minimal installations, and at least 10 GB for installations with a graphical environment.

- Consider leaving a portion of the space in an LVM volume group unallocated. This unallocated space gives you flexibility if your space requirements change but you do not wish to remove data from other volumes. You can also select the **Thin provisioning** device type for the partition to have the unused space handled automatically by the volume.

- The size of an XFS file system can not be reduced - if you need to make a partition or volume with this file system smaller, you must back up your data, destroy the file system, and create a new, smaller one in its place. Therefore, if you expect needing to manipulate your partitioning layout later, you should use the `ext4` file system instead.

- Use Logical Volume Management (LVM) if you anticipate expanding your storage by adding more hard drives after the installation. With LVM, you can create physical volumes on the new drives, and then assign them to any volume group and logical volume as you see fit - for example, you can easily expand your system's **`/home`** (or any other directory residing on a logical volume).

- Creating a BIOS Boot partition or an EFI System Partition may be necessary, depending on your system's firmware, boot drive size, and boot drive disk label. See *Section 5.4.10.6, “Recommended Partitioning Scheme”* for information about these partitions. Note that the graphical installer will not let you create a BIOS Boot or EFI System Partition if your system does *not* require one - in that case, they will be hidden from the menu.

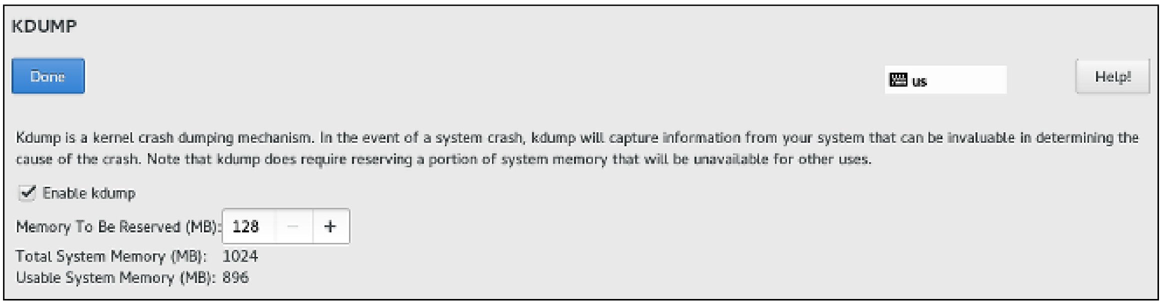
- If you need to make any changes to your storage configuration after the installation, Fedora repositories offer several different tools which can help you do this. If you prefer a command line tool, try *system-storage-manager*.

5.4.11. Kdump

Note

This screen is disabled by default. To enable it during the installation, you must use the **`inst.kdump_addon=on`** option at the boot menu. See *Section 8.2.5, “Advanced Installation Options”* for details, and *Section 4.2, “The Boot Menu”* for instructions on using custom boot options.

Use this screen to select whether or not **Kdump** will be activated on the installed system, and how much memory will be reserved for it if enabled.



Kdump is a kernel crash dumping mechanism. In the event of a system crash, kdump will capture information from your system that can be invaluable in determining the cause of the crash. Note that kdump does require reserving a portion of system memory that will be unavailable for other uses.

☑ Enable kdump

Memory To Be Reserved (MB): 128 − +

Total System Memory (MB): 1024

Usable System Memory (MB): 896

Figure 5.22. Kdump

Kdump is a kernel crash dumping mechanism which, in the event of a system crash, captures the contents of the system memory at the moment of failure. This captured memory can then be analyzed to find the cause of the crash. If **Kdump** is enabled, it must have a small portion of the system's memory (RAM) reserved to itself. This reserved memory will not be accessible to the main kernel.

To enable **Kdump** on the installed system, check **Enabled**. Then, select either **Automatic** or **Manual** memory reservation settings, and if you selected **Manual**, enter the amount of memory to be reserved in megabytes into the **Memory to be reserved** field.

The amount of memory which you should reserve is determined based on your system's architecture (AMD64 and Intel 64 will have different requirements than IBM Power, for example) as well as the total amount of system memory. In most cases, automatic reservation will be satisfactory. If you insist on manual settings, see the *Red Hat Enterprise Linux 7 Kernel Crash Dump Guide*[2] for guidelines. This document also contains more in-depth information about how **Kdump** works, how to configure additional settings, and how to analyze a saved crash dump.

The **Usable System Memory** readout below the reservation input field shows how much memory will be accessible to your main system once your selected amount of RAM is reserved.

Note

Additional settings, such as the location where kernel crash dumps will be saved, can only be configured after the installation using either the `system-config-kdump` graphical interface, or manually in the **`/etc/kdump.conf`** configuration file.

After configuring **Kdump** settings, click **Done** in the top left corner to return to *Section 5.4.2, “Installation Summary”*.

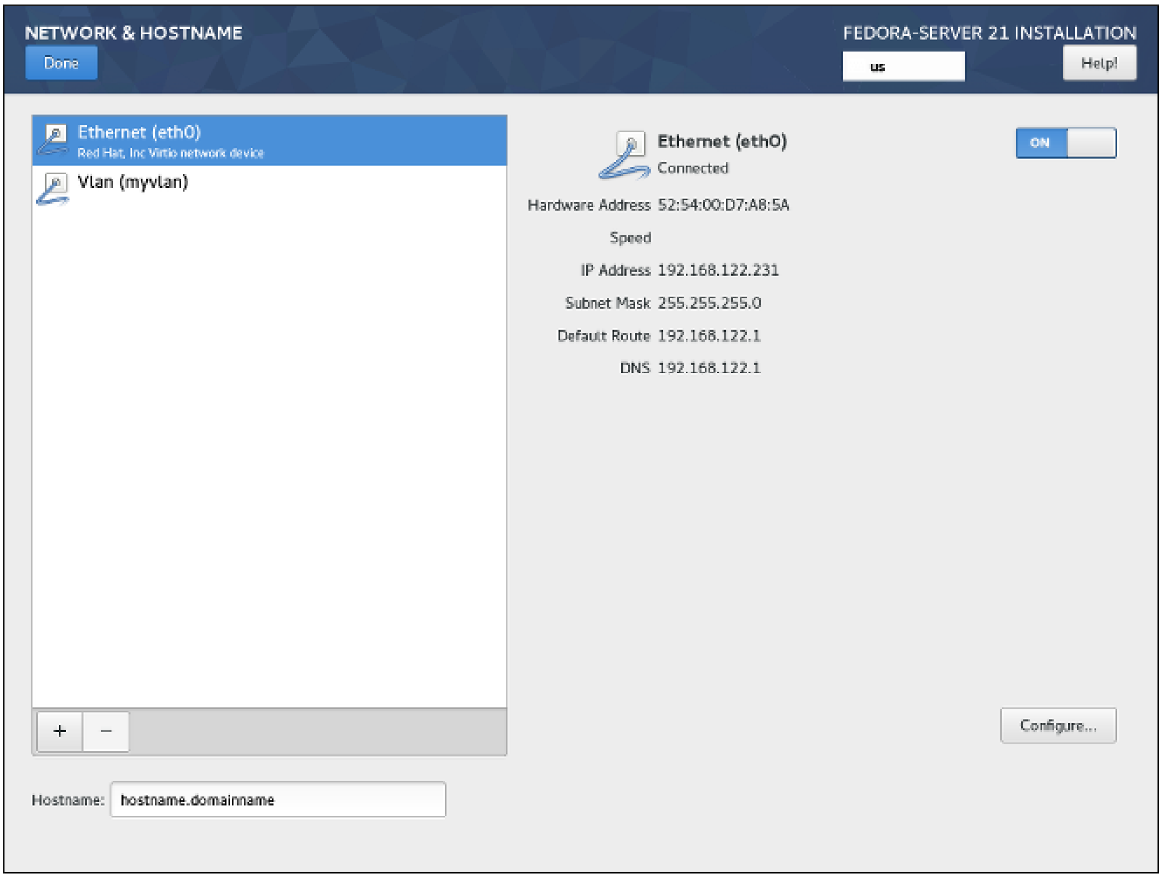
5.4.12. Network & Hostname

The **Network & Hostname** screen is used to configure network interfaces. Options selected here will be available both during the installation (if needed for tasks such as downloading packages from a remote location) and on the installed system.

Note

Network configuration is an expansive topic and many of the options available during the installation are beyond the scope of this document. For detailed information about networking, including both theoretical topics and specific instructions and examples, see the *Fedora Networking Guide*, available at *http://docs.fedoraproject.org/*.

[2] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Kernel_Crash_Dump_Guide/appe-supported-kdump-configurations-and-targets.html#sect-kdump-memory-requirements



Ethernet (eth0)
Connected

Hardware Address 52:54:00:D7:A8:5A
Speed
IP Address 192.168.122.231
Subnet Mask 255.255.255.0
Default Route 192.168.122.1
DNS 192.168.122.1

Hostname: hostname.domainname

Figure 5.23. Network & Hostname

Locally accessible interfaces are automatically detected by the installation program and cannot be manually added or deleted. All detected interfaces are listed on the left side of the screen. Click an interface in the list to display its current configuration (such as IP and DNS address); the details are displayed on the right side of the screen.

Below the list of interfaces are two buttons. Use the **+** button to add a virtual network interface (Team, Bond or VLAN) as described in *Section 5.4.12.1, “Adding a Virtual Network Interface”*. To remove a previously created virtual interface, select it in the list and click the **-** button.

To change settings such as IP addresses, DNS servers, or routing configuration for an existing interface (both virtual and physical), select the interface in the left pane and click **Configure** in the bottom right corner of the screen. Available settings are described in *Section 5.4.12.2, “Editing Network Interface Configuration”*.

Use the **ON**/**OFF** switch in the top right corner to enable or disable the currently selected interface.

Below the list of connections, enter a host name for this computer in the **Hostname** input field. The host name can be either a *fully-qualified domain name* (FQDN) in the format *`hostname.domainname`*, or a short host name with no domain name. Many networks have a *Dynamic Host Configuration Protocol* (`DHCP`) service that automatically supplies connected systems with a domain name; to allow the `DHCP` service to assign the domain name to this machine, only specify the short host name.

5.4.12.1. Adding a Virtual Network Interface

To add a virtual network interface, click the **+** button at the bottom of the interface list. A new window will open, prompting you to select one of the three available types of virtual interfaces:

- **Bond** - NIC (*Network Interface Controller*) Bonding, a method to bind multiple physical network interfaces together into a single bonded channel.

- **Team** - NIC Teaming, a new implementation to aggregate links, designed to provide a small kernel driver to implement the fast handling of packet flows, and various applications to do everything else in user space.
- **Vlan** (*Virtual LAN*) - A method to create multiple distinct broadcast domains which are mutually isolated.

Select the interface type you want to add, and click **Add**. Another dialog window will open, allowing you to edit any available settings for your chosen interface type. For information about available settings, see the respective sections of the *Fedora Networking Guide*, available at *http://docs.fedoraproject.org/*. Basic documentation is also available in *Section 5.4.12.2, "Editing Network Interface Configuration"*.

Note

To access the settings dialog again after you closed it, select the same interface in the list of configured interfaces and click **Configure** in the bottom right corner of the screen. To remove a virtual interface, select it in the list and click the **-** button below.

5.4.12.2. Editing Network Interface Configuration

This section only details the most important settings for a typical wired connection used during installation. Many of the available options do not have to be changed in most installation scenarios and are not carried over to the installed system. Configuration of other types of networks is broadly similar, although the specific configuration parameters may be different. To learn more about network configuration after installation, see the *Fedora Networking Guide*, available at *http://docs.fedoraproject.org/*.

To configure a network connection manually, select that connection in the list on the left side of the screen, and click the **Configure** button. A dialog will appear that allows you to configure the selected connection. The configuration options presented depends on the connection type - the available options will be slightly different depending on whether it is a physical interface (wired or wireless network interface controller) or a virtual interface (Bond, Team or Vlan) which you previously configured in *Section 5.4.12.1, "Adding a Virtual Network Interface"*.. A full description of all configuration settings for all connection types is beyond the scope of this document; see the *Networking Guide* for details.

The most common and useful options in the configuration dialog are:

Enable or disable the connection by default

> In the **General** tab of the configuration dialog, you can select or unselect the **Automatically connect to this network when it is available** check box to allow or disallow this connection to connect by default. When enabled on a wired connection, this means the system will typically connect during startup (unless you unplug the network cable); on a wireless connection, it means that the interface will attempt to connect to any known wireless networks in range.
>
> Additionally, you can allow or disallow all users on the system from connecting to this network using the **All users may connect to this network** option. If you disable this option, only `root` will be able to connect to this network.

> **Note**
>
> It is not possible to only allow a specific user other than `root` to use this interface, because no other users are created at this point during the installation. If you need a connection for a different user, you must configure it after the installation.

Set up static IPv4 or IPv6 settings

By default, both `IPv4` and `IPv6` are set to automatic configuration depending on current network settings. This means that addresses such as the local IP address, DNS address, and other settings will be detected automatically each time the interface connects to a network. In many cases, this is sufficient, but you can also provide static configuration in the **IPv4 Settings** and **IPv6 Settings**, respectively.

To set static network configuration, navigate to one of the settings tabs and select a method other than **Automatic** (for example, **Manual**) from the **Method** drop-down menu. This will enable the **Addresses** field below.

> **Note**
>
> In the **IPv6 Settings** tab, you can also set the method to **Ignore** to disable `IPv6` on this interface.

Then, click **Add** on the right side and add a set of settings: **Address**, **Netmask** (for `IPv4`), **Prefix** (for `IPv6`), and **Gateway**.

The **DNS servers** field accepts one or more IP addresses of DNS servers - for example, **`10.0.0.1,10.0.0.8`**.

The final option in both tabs is **Require IPv*X* addressing for this connection to complete**. Select this option in the `IPv4` tab to only allow this connection if `IPv4` was successful; the same principle applies to this setting in the `IPv6` tab. If this option remains disabled for both `IPv4` and `IPv6`, the interface will be able to connect if configuration succeeds on either IP protocol.

Configure routes

In the **IPv4 Settings** and **IPv4 Settings** tabs, click the **Routes** button in the bottom right corner to configure routing settings for a specific IP protocol on an interface. A new dialog will open, allowing you to **Add** a specific route.

If you confire at least one static route, you can disallow all routes not specifically configured here by enabling the **Ignore automatically obtained routes**.

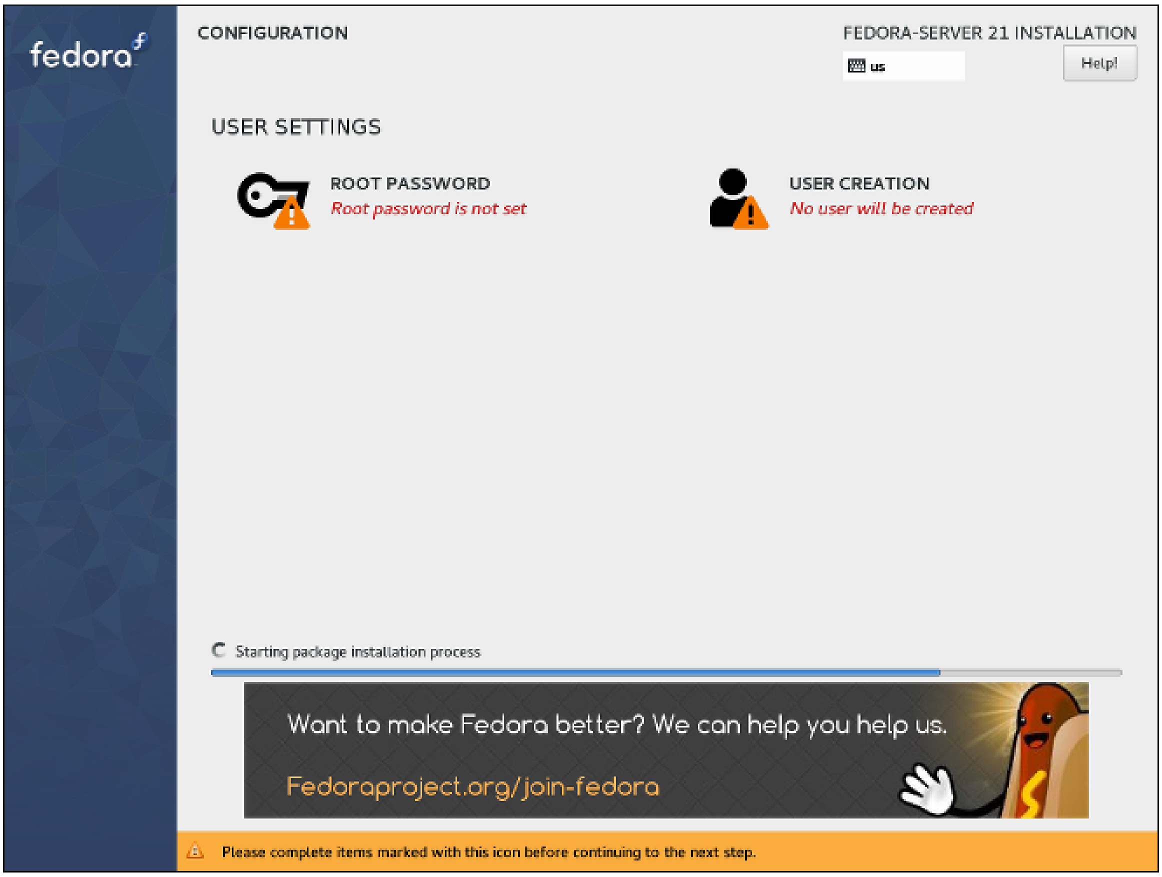
Select **Use this connection only for resources on its network** to prevent this connection from becoming the default route. This option can be selected even if you did not configure any static routes. Enabling this option means that this route will only be used when necessary to access certain resources, such as intranet pages which require a local or VPN connection. Another (default) route will be used for publicly available resources if possible. Note that unlike the additional routes configured in this dialog, this setting will be transferred to the installed system. Also note that this option is only useful when more than one interface is configured.

When you finish configuring the interface's routing settings, click **OK** to return to the configuration dialog.

Once you finish configuring the interface, click **Save** in the configuration window's bottom right corner to save your settings and return to *Section 5.4.12, “Network & Hostname”*.

5.4.13. Configuration and Installation Progress

The **Configuration** screen is displayed after you finish configuring all required items in *Section 5.4.2, “Installation Summary”* and press the **Begin installation** button. After this point, the installation process actually starts and changes are being made to your selected disks. It is not possible to go back to the **Installation Summary** and change any settings configured there; if you need to do so, you must wait for the installation process to finish, reboot your system, log in and change your settings on the installed system.


fedora
CONFIGURATION
FEDORA-SERVER 21 INSTALLATION
us
Help!
USER SETTINGS
ROOT PASSWORD
USER CREATION
Want to make Fedora better? We can help you help us.
Fedoraproject.org/join-fedora

Root password is not set

No user will be created

Starting package installation process

Please complete items marked with this icon before continuing to the next step.

Figure 5.24. Configuration and Installation Progress

In the top part of the screen, two additional options are available, allowing you to configure authentication settings - the `root` password and an additional user account. Configuring the `root` password is required to finish the installation; creating a user account can be skipped. If you do so, you will be prompted to create a non-root user account after the installation finishes and your system reboots.

The bottom of the screen shows a progress bar and a message informing you of the current progress of the installation. When the installation finishes and the `root` password has been set, you can press the **Finish configuration** button to reboot your computer and log in to your newly installed Fedora system.

Important

Before you finish the installation and reboot, either remove the media (CD, DVD or a USB drive) which you used to start the installation, or make sure that your system tries to boot from the hard drive before trying removable media. Otherwise, your computer will start the installer again instead of the installed system.

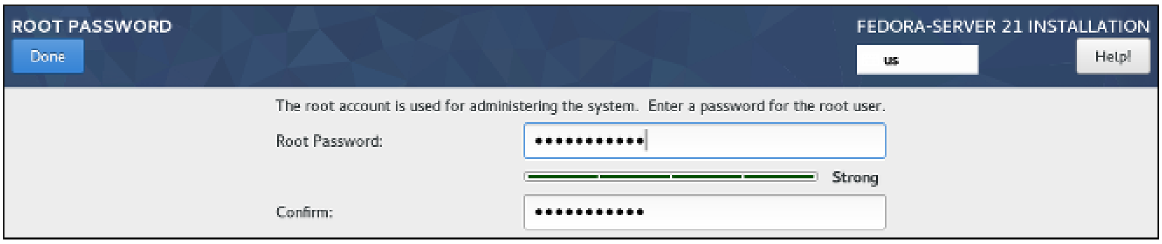
5.4.14. Root Password

The **Root Password** screen is used to configure the `root` password for your system. This password will be used to log into the administrator (also known as superuser) account, which is used for system administration tasks such as installing and updating software packages and changing system-wide configuration such as network and firewall settings, storage options and adding or modifying users, groups and file permissions.

The `root` account will always be created during the installation. However, you should always also create a normal user accont in *Section 5.4.15, “Create User”* and use that account to log in to the system. Only switch to the administrator account only when you need to perform a task which requires administrator access.

Warning

The `root` account has complete control over the system. If an unauthorized person gains access to this account, access or delete all users' personal files or otherwise exploit the machine for their own nefarious purposes. See the *Fedora Security Guide*, available at *http://docs.fedoraproject.org/*, for detailed documentation about account security and guidelines for choosing a strong password.



The root account is used for administering the system. Enter a password for the root user.

Root Password: ●●●●●●●●●●●●

Strong

Confirm: ●●●●●●●●●●●●

Figure 5.25. Root Password

Once you choose a strong password, enter it in the **Root Password** field. The characters you write will be displayed as dots for security. Then, type the same password into the **Confirm** field to ensure you entered it properly. Both entered passwords must be the same.

As you enter the password, it will be evaluated and the installer will determine the password's strength. If the installer considers your password weak, a message will appear at the bottom of the screen, explaining which aspect of your chosen password is considered insuficient. For example:

```
The password you have provided is weak: The password is shorter than 5 characters.
```

If a message similar to the above appears, it is highly recommended to choose a different, stronger password.

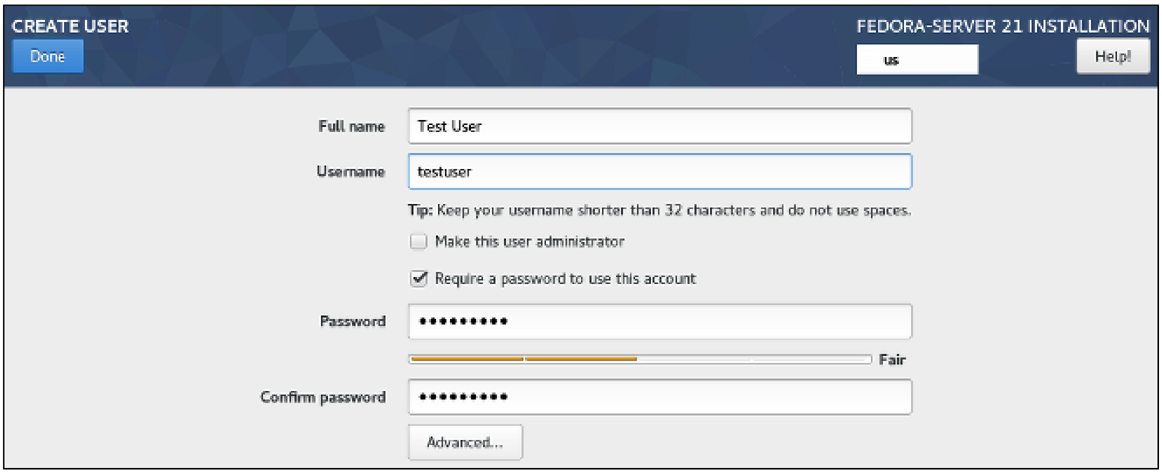
Once you configure the superuser password, click **Done** in the top left corner to return to *Section 5.4.13, “Configuration and Installation Progress”*. If you selected a weak password, you must press the button twice.

5.4.15. Create User

The **Create User** screen can be used to create and configure one normal (non-`root`) user account during the installation. Only one user account can be configured here; if you require more accounts, wait until the installation completes, reboot your system and then create other accounts using either the **`useradd`** command in a terminal, or a graphical utility for managing user accounts (for example, the **Users** screen in the **Settings** menu in **GNOME**).

Important

Creating a normal user account is not required to finish the installation; however, it is highly recommended. If you do not create one, you will have to log in to the system as `root` directly, which is *not* recommended.



Full name: Test User

Username: testuser

Tip: Keep your username shorter than 32 characters and do not use spaces.

□ Make this user administrator

☑ Require a password to use this account

Password: •••••••••

Fair

Confirm password: •••••••••

Figure 5.26. Create User

To configure a user account, fill out the **Full name** (for example, **`John Smith`**) and **Username** (for example, **`jsmith`**). The **username** will used to log in from a command line; if you install a graphical environment, then your graphical login manager will use **Full name**.

Make sure that the **Require a password to use this account** check box is enabled, enter a password you want to use for this account into the **Password** field. The characters you write will be displayed as dots for security. Then, repeat the same password in the **Confirm password** field below to ensure you entered it properly. Both entered passwords must be the same.

As you enter the password, it will be evaluated and the installer will determine the password's strength. If the installer considers your password weak, a message will appear at the bottom of the screen, explaining which aspect of your chosen password is considered insuficient. For example:

```
The password you have provided is weak: The password is shorter than 5 characters.
```

If a message similar to the above appears, it is highly recommended to choose a different, stronger password.

The **Make this user administrator** check box gives the user you are creatng administrative rights (by adding this user into the `wheel` group), allowing this user to use the **sudo** command to perform tasks normally only available to `root` using only the user password, instead of the `root` password. This may be more convenient, but can also pose a security risk.

Warning

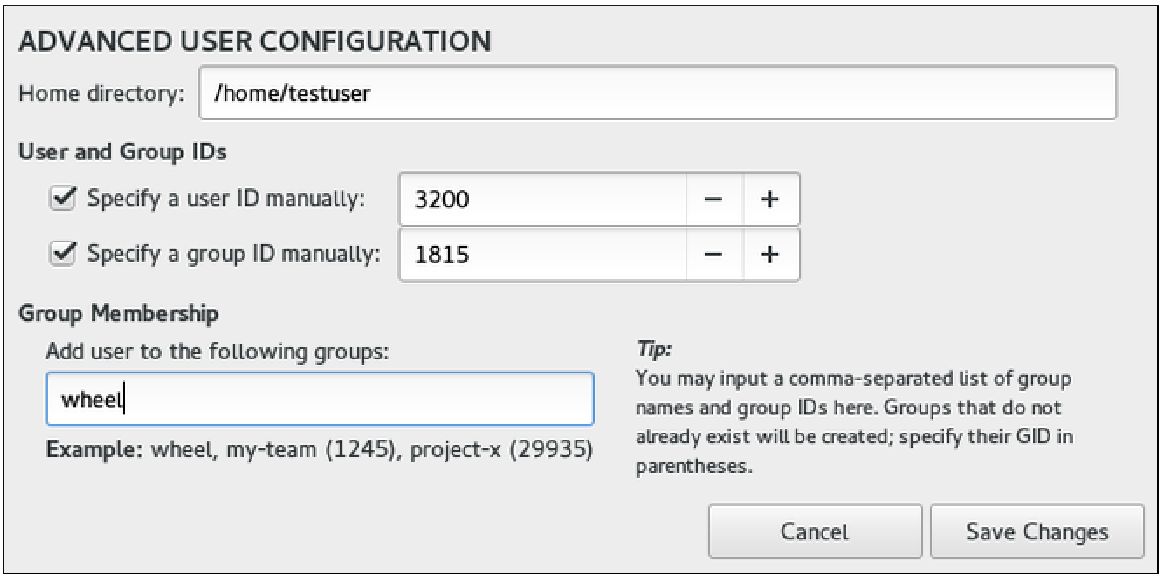
If you give administrator privileges to an user, make sure that the account is protected by a strong password. Never give an user administrator privileges without requiring any password for the account.

See the *Fedora Security Guide*, available at the *http://docs.fedoraproject.org/*, for detailed information about password security including guidelines for selecting strong passwords.

To additional settings, click the **Advanced** button below the password input fields. A new dialog window titled **Advanced User Configuration** will open. This dialog is described in the following section.

5.4.15.1. Advanced User Configuration

The **Advanced User Configuration** dialog allows you to change the following settings for the new user account.



ADVANCED USER CONFIGURATION

Home directory: /home/testuser

User and Group IDs

☑ Specify a user ID manually: 3200

☑ Specify a group ID manually: 1815

Group Membership

Add user to the following groups: wheel

Example: wheel, my-team (1245), project-x (29935)

Tip: You may input a comma-separated list of group names and group IDs here. Groups that do not already exist will be created; specify their GID in parentheses.

Figure 5.27. Advanced User Configuration

- The user's **Home directory** (**`/home/`*`username`*** by default).

- The user's ID (UID). The default value is 1000. UIDs 0-999 are reserved by the system so they can not be assigned to an user.

- The the group ID (GID) of the user's default group. The default group name will be the same as the user name, and its default GID is 1000. GIDs 0-999 are reserved by the system so they can not be assigned to an user's group.

- The user's group membership. The user account you are creating will always have one default group membership (the user's default group with an ID set in the **Specify a group ID manually** field). In the **Group Membership** field, you can specify additional groups as a comma-separated list. Groups which do not already exist will be created; you can specify custom GIDs for them in parentheses. If you do not specify a custom GID for a new group, it will be assigned automatically.

After the Installation

This chapter describes the initial setup tools which open after you finish the installation, reboot the system, and log in for the first time.

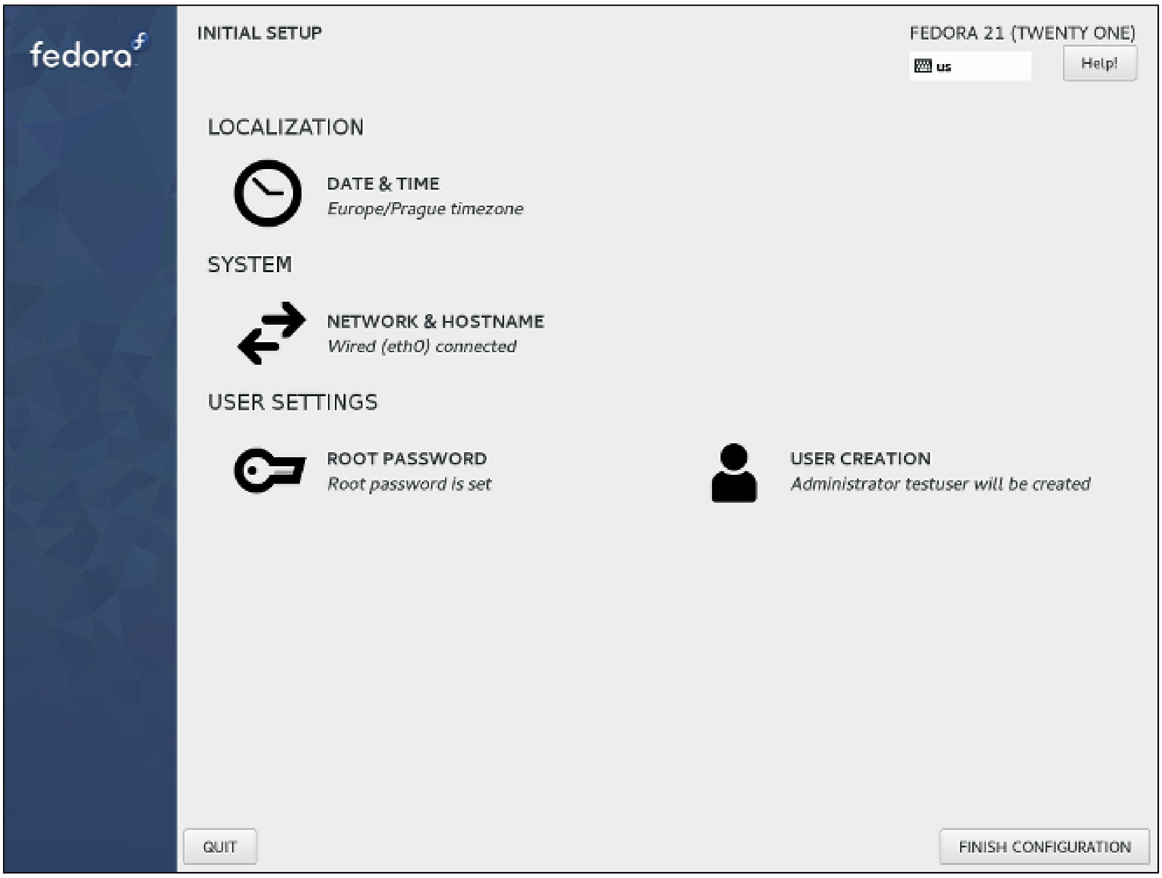
Two separate configuration tools are part of Fedora. If you installed the **GNOME** desktop environment (the default Fedora environment, installed for example as part of the Fedora Workstation flavor), the **GNOME Initial Setup** utility will open; for all other graphical environments, the **Initial Setup** tool serves the same purpose.

Both of these tools will allow you to configure user-specific settings which have not been configured during the installation. Some of the settings are available both in the graphical installer (or as Kickstart commands) as well as in the initial-setup utilities; in that case, they will only be displayed after the first reboot if you skipped them during the installation. This means that, for example, if you configured a user account before the installation finished, you will not be prompted to do so again after the installation completes.

6.1. Initial Setup

The **Initial Setup** screen is shown after the installation finishes and the installed system boots for the first time, before the first login, assuming the following conditions have been met:

- Your software selection contained a graphical environment (for example, you installed the system using a Fedora Workstation live image).
- The graphical environment you have installed is *not* **GNOME Desktop Environment** (it provides its own initial setup utility which is described in *Section 6.2, “GNOME Initial Setup”*).
- The *initial-setup* package has been installed.
- You have not configured every screen available in the graphical installer.



LOCALIZATION

DATE & TIME
Europe/Prague timezone

SYSTEM

NETWORK & HOSTNAME
Wired (eth0) connected

USER SETTINGS

ROOT PASSWORD
Root password is set

USER CREATION
Administrator testuser will be created

Figure 6.1. Initial Setup

Initial Setup allows you to configure several system settings. All of these can also be configured during the installation, but they are not required to finish it. Only the settings which have *not* been configured during the installation will be shown; for example, if you did not create a non-`root` user account during the installation, **Initial Setup** will start after the first reboot, and you will be able to configure one. If you configured all available options during the instalaltion, **Initial Setup** will not be displayed at all.

To configure any aspect of the system available in the utility, click any of the links available in the main window (for example, **Create User**). Each link leads to a separate screen with separate options; these screens are the same ones which were available during the graphical installation. If you need help, press **Help** in the top left corner to open a new window containing directions for that screen.

After you finish configuring any screen, press the **Done** button in the top left corner to return to the main **Initial Setup** menu. After you finish configuring all settings, click **Finish Configuration** in the bottom right corner to save all configured settings. The configuration utility will close and you will be able to log in to the system.

Note

It is possible to configure **Initial Setup** to display all available options, even if they have been already configured during the installation. To do so, you must use a Kickstart file at the start of the installation, and this file must contain the following command:

```
firstboot --enable --reconfig
```

The `--reconfig` option specifies that all options should be displayed. See *Chapter 9, Automating the Installation with Kickstart* for information about Kickstart installations.

Note

Normally, it is not possible to return to **Initial Setup** after you close it and log in to the system. You can make it display again (after the next reboot, before a login prompt is displayed), by executing the following command as `root`:

```
# systemctl enable initial-setup-graphical.service
```

Then, reboot your system.

6.2. GNOME Initial Setup

The **GNOME Initial Setup** utility will be displayed after you log in for the first time with any user account, as long as your first login is in the **GNOME** desktop environment.

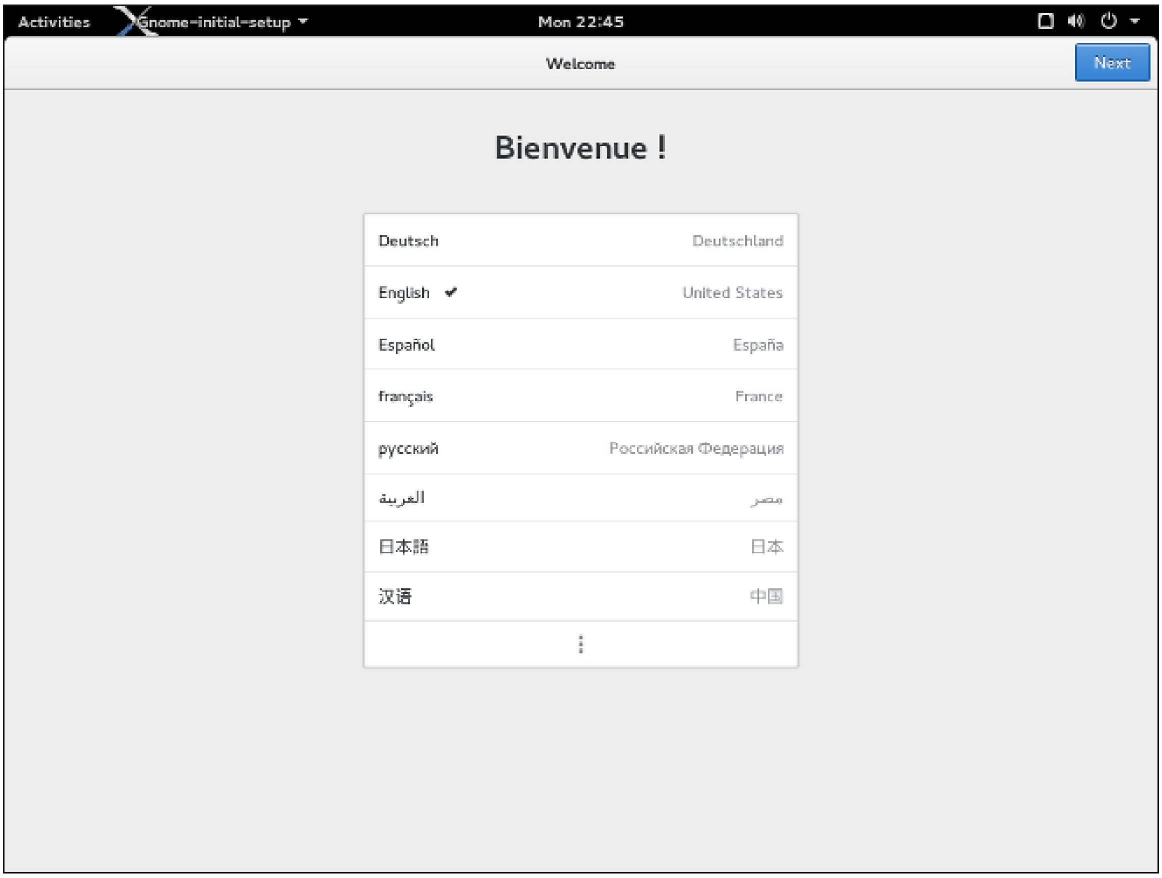


Figure 6.2. GNOME Initial Setup

This utility allows you to configure several user-specific settings, such as system language or the default keyboard layout. Use the **Next** and **Previous** buttons at the top of the window to navigate between screens. Some screens (**Configure online accounts** in particular) are optional and can be skipped using the **Skip** button in the top right corner.

After you finish configuring all screens, click **Start using Fedora** to exit the utility. After that, the **Getting Started** page from GNOME help will be displayed, and you can either browse the help content, or close it and start using your desktop.

6.3. Common Post-installation Tasks

After you finish the installation and go through one of the initial setup utilities described above, your system is ready for use. However, other administrative tasks not covered by the installer or the initial setup utilities may still be necessary, depending on how you plan to use your system. The list below describes some of the more common tasks usually performed immediately after a new installation.

Recover a lost `root` password

The `root` password, which is configured during the installation, is required for accessing the system as the `root` user. Without this password you will not be able to configure your system or install additional software. If you lost or forgot your `root` password, you can reset it by following the steps described in *Section 7.4.1, “Resetting the Root Password”*.

Configure the network

In most cases, network access is configured during the installation process, either in the **Anaconda** installation program (see *Section 5.4.12, “Network & Hostname”*) or in a Kickstart file (see *Section A.3.2, “network (optional) - Configure Network Interfaces”*). For information on configuring the network after the installation, see the *Fedora Networking Guide*, available at *http://docs.fedoraproject.org/*.

Perform an initial system update

If you installed Fedora from a full installation ISO image (such as the Workstation live image or full Server DVD), it is recommended to perform an initial system update, because your installation media likely contained older versions of software packages. During the initial update, all installed packages are updated to their latest available versions. Updates to packages provide security fixes, bug fixes and enhancements.

The **DNF** package manager is used for updating installed packages. For more information about updating your system with **DNF**, see the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*.

Install driver updates

Usually, drivers for system devices are already supported in the kernel provided by Red Hat Enterprise Linux. However, occasionally, support for devices that have been released recently could be missing. In these cases, a driver update enabling your device may be available.

Devices necessary to complete the installation can have driver updates provided before the installation begins. If a device is missing a driver, but it is not essential during the installation, it is recommended to wait until after the installation completes, and install additional drivers afterwards. For instructions on installing and enabling additional drivers on the installed system using **RPM** and **DNF**, see the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*.

Configure additional repositories

New software is installed from *package repositories*. Package repositories are organized sets of software and metadata that can be accessed by the **DNF** package manager. By default, several update repositories provided by Fedora are configured automatically and you can install updates and additional software from those. However, if you want to set up additional repositories, for example ones containing your own software, some extra steps are needed.

For information about configuring additional software repositories, see the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*.

Install additional packages

You can control which packages will be installed by selecting an environment in the `Software Selection` dialog in the graphical installation (see *Section 5.4.7, “Software Selection”*). This dialog does not provide a way to choose individual packages, only predefined sets. However, you can use the **DNF** package manager to install additional packages after the installation. See the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*, for more information.

Switch to a graphical login

Depending on the options you chose during the installation process, it is possible that your system does not have a graphical interface, instead offering only a text-based prompt. If this is the case and you wish to enable a graphical desktop after the installation, you must install the **X Window System** and your preferred desktop environment (either **GNOME** or **KDE**).

As with all other software, these packages can be installed using the **DNF** package manager. For information about using **DNF** to install new packages, see the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*. For information on how to enable graphical login by default, see *Section 7.4.4, “Booting into a Graphical Environment”*.

Troubleshooting

This chapter offers some pointers on how to get help when something goes wrong. It also discusses some common installation problems and their solutions.

7.1. Getting Help

There are many places on the internet which can help you when you encounter a problem not described in this chapter: discussion boards, blogs, IRC, and more. Some of the more popular places where you can find help include:

- *Ask Fedora*[1] - Fedora's knowledge base, available in multiple languages
- The #fedora IRC channel on *FreeNode*[2] - one of the main IRC channels used by Fedora users, English only
- *Fedora Project Wiki*[3] - the official wiki for Fedora Project
- *Stack Exchange*[4] - an English language Q&A board, not specific to Fedora

> **Note**
>
> The above list is by no means complete - you can find help in many other places as well. Additional information about available resources such as IRC channels and mailing lists is available at *https://fedoraproject.org/wiki/Communicating_and_getting_help*.

Before you open a new discussion or ask anyone for help on IRC, you should always do some research on your own. If you are encountering an issue, there is usually a good chance that someone else ran into the same problem before you and published a solution somewhere. Opening a discussion about something already explained elsewhere, or asking a common question which has been answered many times before, is not likely to result in a friendly, constructive response.

When you ask for help troubleshooting problems related to the installation, you may be asked to provide log files generated by the installer. The sections below explain which files are generated, what their contents are, and how to transfer them from the installation system.

7.1.1. Log Files Generated During the Installation

For debugging purposes, **Anaconda** logs installation actions into files in the **`/tmp`** directory. These files are listed in the following table.

Table 7.1. Log Files and Their Contents

Log file	Contents
`/tmp/anaconda.log`	general **Anaconda** messages
`/tmp/program.log`	all external programs run during the installation

[1] https://ask.fedoraproject.org/en/questions/
[2] http://freenode.net/
[3] https://fedoraproject.org/wiki/
[4] http://unix.stackexchange.com/

Log file	Contents
`/tmp/storage.log`	extensive storage module information
`/tmp/packaging.log`	**dnf** and **rpm** package installation messages
`/tmp/syslog`	hardware-related system messages

If the installation fails, the messages from these files are consolidated into **`/tmp/anaconda-tb-`*`identifier`***, where *`identifier`* is a random string.

7.1.2. Transferring Log Files from the Installation System

All of the files described in *Section 7.1.1, “Log Files Generated During the Installation”* reside in the installation program's RAM disk, which means they are not saved permamently and will be lost once the system is powered down. To store them permanently, copy those files to another system on the network using **scp** on the system running the installation program, or copy them to a mounted storage device (such as an USB flash drive). Details on how to transfer the log files are below. Note that if you use an USB flash drive or other removable media, you should make sure to back up any data on it before starting the procedure.

7.1.2.1. Transferring Log Files Onto a USB Drive

1. On the system you are installing, press **Ctrl**+**Alt**+**F2** to access a shell prompt. You will be logged into a root account and you will have access to the installation program's temporary file system.

2. Connect a USB flash drive to the system and execute the **dmesg** command. A log detailing all recent events will be displayed. At the bottom of this log, you will see a set of messages caused by the USB flash drive you just connected. It will look like a set of lines similar to the following:

```
[ 170.171135] sd 5:0:0:0: [sdb] Attached SCSI removable disk
```

   Note the name of the connected device - in the above example, it is **sdb**.

3. Go to the **/mnt** directory and once there, create new directory which will serve as the mount target for the USB drive. The name of the directory does not matter; this example uses the name **usb**.

```
# mkdir usb
```

4. Mount the USB flash drive onto the newly created directory. Note that in most cases, you do not want to mount the whole drive, but a partition on it. Therefore, do not use the name **sdb** - use the name of the partition you want to write the log files to. In this example, the name **sdb1** is used.

```
# mount /dev/sdb1 /mnt/usb
```

   You can now verify that you mounted the correct device and partition by accessing it and listing its contents - the list should match what you expect to be on the drive.

```
# cd /mnt/usb
```

```
# ls
```

5. Copy the log files to the mounted device.

   ```
   # cp /tmp/*log /mnt/usb
   ```

6. Unmount the USB flash drive. If you get an error message saying that the target is busy, change your working directory to outside the mount (for example, **`/`**).

   ```
   # umount /mnt/usb
   ```

The log files from the installation are now saved on the USB flash drive.

7.1.2.2. Transferring Log Files Over the Network

1. On the system you are installing, press **Ctrl**+**Alt**+**F2** to access a shell prompt. You will be logged into a root account and you will have access to the installation program's temporary file system.

2. Switch to the **`/tmp`** directory where the log files are located:

   ```
   # cd /tmp
   ```

3. Copy the log files onto another system on the network using the **`scp`** command:

   ```
   # scp *log user@address:path
   ```

   Replace *`user`* with a valid user name on the target system, *`address`* with the target system's address or host name, and *`path`* with the path to the directory you wish to save the log files into. For example, if you want to log in as `john` to a system with an IP address of `192.168.0.122` and place the log files into the **`/home/john/logs/`** directory on that system, the command will have the following form:

   ```
   # scp *log john@192.168.0.122:/home/john/logs/
   ```

   When connecting to the target system for the first time, you may encounter a message similar to the following:

   ```
   The authenticity of host '192.168.0.122 (192.168.0.122)' can't be established.
   ECDSA key fingerprint is a4:60:76:eb:b2:d0:aa:23:af:3d:59:5c:de:bb:c4:42.
   Are you sure you want to continue connecting (yes/no)?
   ```

   Type **`yes`** and press **Enter** to continue. Then, provide a valid password when prompted. The files will start transferring to the specified directory on the target system.

The log files from the installation are now permanently saved on the target system and available for review.

7.2. Trouble Beginning the Installation

7.2.1. Problems with Booting into the Graphical Installation

Systems with some video cards have trouble booting into the graphical installation program. If the installation program does not run using its default settings, it attempts to run in a lower resolution mode. If that still fails, the installation program attempts to run in text mode.

There are several possible solutions to display issues, most of which involve specifying custom boot options. For more information, see *Section 8.1, “Configuring the Installation System at the Boot Menu”*.

Use the basic graphics mode

> You can attempt to perform the installation using the basic graphics driver. To do this, edit the installation program's boot options and append **`inst.xdriver=vesa`** at the end of the command line.

Specify the display resolution manually

> If the installation program fails to detect your screen resolution, you can override the automatic detection and specify it manually. To do this, append the **`inst.resolution=`*`x`*** option at the boot menu, where *x* is your display's resolution (for example, **`1024x768`**).

Use an alternate video driver

> You can also attempt to specify a custom video driver, overriding the installation program's automatic detection. To specify a driver, use the **`inst.xdriver=`*`x`*** option, where *x* is the device driver you want to use (for example, **`nouveau`**).
>
> **Note**
>
> If specifying a custom video driver solves your problem, you should report it as a bug at *https://bugzilla.redhat.com* under the **anaconda** component. **Anaconda** should be able to detect your hardware automatically and use the appropriate driver without your intervention.

Perform the installation using VNC

> If the above options fail, you can use a separate system to access the graphical installation over the network, using the *Virtual Network Computing* (VNC) protocol. For details on installing using VNC, see *Chapter 11, Installing Using VNC*.

7.2.2. Serial Console Not Detected

In some cases, attempting to install in text mode using a serial console will result in no output on the console. This happens on systems which have a graphics card, but no monitor connected. If **Anaconda** detects a graphics card, it will attempt to use it for a display, even if no display is connected.

If you want to perform a text-based installation on a serial console, use the **`inst.text`** and **`console=`** boot options. See *Chapter 8, Boot Options* for more details.

7.3. Trouble During the Installation

7.3.1. No Disks Detected

In the **Installation Destination** screen, the following error message may appear at the bottom: **No disks detected. Please shut down the computer, connect at least one disk, and restart to complete installation**.

The message indicates that **Anaconda** did not find any writable storage devices to install to. In that case, first make sure that your system does have at least one storage device attached.

If your system uses a hardware RAID controller, verify that the controller is properly configured and working. See your controller's documentation for instructions.

If you are installing into one or more iSCSI devices and there is no local storage present on the system, make sure that all required LUNs (*Logical Unit Numbers*) are being presented to the appropriate HBA (*Host Bus Adapter*).

If you made sure you have a connected and properly configured storage device and the message still appears after you reboot the system and start the installation again, it means that the installation program failed to detect the storage. In most cases this message appears when you attempt to install on an SCSI device which has not been recognized by the installation program.

7.4. Problems After Installation

7.4.1. Resetting the Root Password

If you lost the root password to the system and you have access to the boot loader, you can reset the password by editing the GRUB2 configuration.

Procedure 7.1. Resetting the Root Password

1. Boot your system and wait until the GRUB2 menu appears.

2. In the boot loader menu, highlight any entry and press **`e`** to edit it.

3. Find the line beginning with **`linux`**. At the end of this line, append the following:

   ```
   init=/bin/sh
   ```

4. Press **`F10`** or **`Ctrl`**+**`X`** to boot the system using the options you just edited.

   Once the system boots, you will be presented with a shell prompt without having to enter any user name or password:

   ```
   sh-4.2#
   ```

5. Load the installed SELinux policy:

   ```
   sh-4.2# /usr/sbin/load_policy -i
   ```

6. Execute the following command to remount your root partition:

   ```
   sh4.2# mount -o remount,rw /
   ```

7. Reset the root password:

   ```
   sh4.2# passwd root
   ```

   When prompted to, enter your new root password and confirm by pressing the **Enter** key. Enter the password for the second time to make sure you typed it correctly and confirm with **Enter** again. If both passwords match, a message informing you of a successful root password change will appear.

8. Remount the root partition again, this time as read-only:

   ```
   sh4.2# mount -o remount,ro /
   ```

9. Reboot the system. From now on, you will be able to log in as the root user using the new password set up during this procedure.

7.4.2. Are You Unable to Boot With Your RAID Card?

If you have performed an installation and cannot boot your system properly, you may need to reinstall and partition your system's storage differently.

Some BIOS types do not support booting from RAID cards. After you finish the installation and reboot the system for the first time, a text-based screen showing the boot loader prompt (for example, `grub>`) and a flashing cursor may be all that appears. If this is the case, you must repartition your system and move your **/boot** partition and the boot loader outside the RAID array. The **/boot** partition and the boot loader must be on the same drive.

Once these changes have been made, you should be able to finish your installation and boot the system properly. For more information about partitioning, see *Section 5.4.8, “Installation Destination”*.

7.4.3. Trouble With the Graphical Boot Sequence

After you finish the installation and reboot your system for the first time, it is possible that the system stops responding during the graphical boot sequence, requiring a reset. In this case, the boot loader is displayed successfully, but selecting any entry and attempting to boot the system results in a halt. This usually means a problem with the graphical boot sequence; to solve this issue, you must disable graphical boot. To do this, temporarily alter the setting at boot time before changing it permanently.

Procedure 7.2. Disabling Graphical Boot Temporarily

1. Start your computer and wait until the boot loader menu appears. If you set your boot loader timeout period to 0, hold down the **Esc** key to access it.

2. When the boot loader menu appears, use your cursor keys to highlight the entry you want to boot and press the **e** key to edit this entry's options.

3. In the list of options, find the kernel line - that is, the line beginning with the keyword **linux** (or, in some cases, **linux16** or **linuxefi**). On this line, locate the **rhgb** option and delete it. The option may not be immediately visible; use the cursor keys to scroll up and down.

4. Press **F10** or **Ctrl**+**X** to boot your system with the edited options.

If the system started successfully, you can log in normally. Then you will need to disable the graphical boot permanently - otherwise you will have to perform the previous procedure every time the system boots. To permanently change boot options, do the following.

Procedure 7.3. Disabling Graphical Boot Permanently

1. Log in to the `root` account using the **`su -`** command:

   ```
   $ su -
   ```

2. Open the **`/etc/default/grub`** configuration file using a plain text editor such as **vim**.

3. Within the **`grub`** file, locate the line beginning with **`GRUB_CMDLINE_LINUX`**. The line should look similar to the following:

   ```
   GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/root rd.md=0 rd.dm=0 vconsole.keymap=us $([ -
   x /usr/sbin/rhcrashkernel-param ] && /usr/sbin/rhcrashkernel-param || :) rd.luks=0
    vconsole.font=latarcyrheb-sun16 rd.lvm.lv=vg_rhel/swap rhgb quiet"
   ```

   On this line, delete the **`rhgb`** option.

4. Save the edited configuration file.

5. Refresh the boot loader configuration by executing the following command:

   ```
   # grub2-mkconfig --output=/boot/grub2/grub.cfg
   ```

After you finish this procedure, you can reboot your computer. Fedora will not use the graphical boot sequence any more. If you wish to enable graphical boot, follow the same procedure, add the **`rhgb`** option to the **`GRUB_CMDLINE_LINUX`** line in the **`/etc/default/grub`** file and refresh the boot loader configuration again using the **`grub2-mkconfig`** command.

See the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*, for more information about working with the **GRUB2** boot loader.

7.4.4. Booting into a Graphical Environment

If you have installed the **X Window System** and a desktop environment such as **GNOME**, but are not seeing a graphical desktop environment once you log into your system, you can start it manually using the **`startx`** command. Note, however, that this is just a one-time fix and does not change the log in process for future log ins.

To set up your system so that you can log in at a graphical login screen, you must change the default **systemd** target to **`graphical.target`**. When you are finished, reboot the computer. You will presented with a graphical login prompt after the system restarts.

Procedure 7.4. Setting Graphical Login as Default

1. Open a shell prompt. If you are in your user account, become root by typing the **`su -`** command.

2. Change the default target to `graphical.target`. To do this, execute the following command:

   ```
   # systemctl set-default graphical.target
   ```

Graphical login is now enabled by default - you will be presented with a graphical login prompt after the next reboot. If you want to reverse this change and keep using the text-based login prompt, execute the following command as `root`:

```
# systemctl set-default multi-user.target
```

For more information about targets in **systemd**, see the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*.

7.4.5. No Graphical User Interface Present

If you are having trouble getting **X** (the **X Window System**) to start, it is possible that it has not been installed. Some of the pre-set base environments you can select during the installation, such as **Minimal install** or **Web Server**, do not include a graphical interface - it has to be installed manually.

If you want **X**, you can install the necessary packages after the installation using the **DNF** package manager. For example, to install **GNOME**, use **`dnf install gnome-shell`** as `root`.

7.4.6. X Server Crashing After User Logs In

If you are having trouble with the **X** server crashing when a user logs in, one or more of your file systems may be full (or nearly full). To verify that this is the problem you are experiencing, execute the following command:

```
$ df -h
```

The output will help you diagnose which partition is full - in most cases, the problem will be on the **`/home`** partition. A sample output of the **`df`** command may look similar to the following:

```

Filesystem                                  Size  Used Avail Use% Mounted on
/dev/mapper/vg_rhel-root                     20G  6.0G   13G  32% /
devtmpfs                                    1.8G     0  1.8G   0% /dev
tmpfs                                       1.8G  2.7M  1.8G   1% /dev/shm
tmpfs                                       1.8G 1012K  1.8G   1% /run
tmpfs                                       1.8G     0  1.8G   0% /sys/fs/cgroup
tmpfs                                       1.8G  2.6M  1.8G   1% /tmp
/dev/sda1                                   976M  150M  760M  17% /boot
/dev/dm-4                                    90G   90G     0 100% /home
```

In the above example, you can see that the **`/home`** partition is full, which causes the crash. You can make some room on the partition by removing unneeded files. After you free up some disk space, start **X** using the **`startx`** command.

For additional information about **`df`** and an explanation of the options available (such as the **`-h`** option used in this example), see the `df(1)` man page.

7.4.7. Is Your RAM Not Being Recognized?

In some cases the kernel does not recognize all of your memory (RAM), which causes the system to use less memory than is installed. You can find out how much RAM is being utilized using the **`free -m`** command. If the displayed total amount of memory does not match your expectations, it is likely that at least one of your memory modules is faulty. On BIOS-based systems, you can use the **Memtest86+** utility to test your system's memory - see *Section 8.5.1, “Loading the Memory (RAM) Testing Mode”* for details.

Note

If you have 4GB or more memory installed, but Fedora only shows around 3.5GB or 3.7GB, you have probably installed a 32-bit version of Fedora on a 64bit kernel. For modern systems, use the 64-bit (x86_64) version.

Some hardware configurations have a part of the system's RAM reserved and unavailable to the main system. Notably, laptop computers with integrated graphics cards will reserve some memory for the GPU. For example, a laptop with 4 GB of RAM and an integrated Intel graphics card will show only roughly 3.7 GB of available memory, even with a 64-bit system.

Additionally, the **kdump** crash kernel dumping mechanism reserves some memory for the secondary kernel used in case of the primary kernel crashing. This reserved memory will also not be displayed as available when using the **`free`** command. For details about **kdump** and its memory requirements, see the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*.

If you made sure that your memory does not have any issues, you can try and set the amount of memory manually using the **`mem=`** kernel option.

Procedure 7.5. Configuring the Memory Manually

1. Start your computer and wait until the boot loader menu appears. If you set your boot loader timeout period to 0, hold down the **`Esc`** key to access it.

2. When the boot loader menu appears, use your cursor keys to highlight the entry you want to boot and press the **`e`** key to edit this entry's options.

3. In the list of options, find the kernel line - that is, the line beginning with the keyword **`linux`** (or, in some cases, **`linux16`**). Append the following option to the end of this line:

   ```
   mem=xxM
   ```

   Replace *xx* with the amount of RAM you have in megabytes.

4. Press **`F10`** or **`Ctrl`**+**`X`** to boot your system with the edited options.

5. Wait for the system to boot and log in. Then, open a command line and execute the **`free -m`** command again. If total amount of RAM displayed by the command matches your expectations, append the following to the line beginning with **`GRUB_CMDLINE_LINUX`** in the **`/etc/default/grub`** file to make the change permanent:

   ```
   mem=xxM
   ```

   Replace *xx* with the amount of RAM you have in megabytes.

6. After you updated the file and saved it, refresh the boot loader configuration so that the change will take effect. Run the following command with root privileges:

   ```
   # grub2-mkconfig --output=/boot/grub2/grub.cfg
   ```

In **`/etc/default/grub`**, the above example would look similar to the following:

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release.*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/root vconsole.font=latarcyrheb-sun16 rd.lvm.lv=rhel/
swap $([ -x /usr/sbin/rhcrashkernel.param ] && /usr/sbin/rhcrashkernel-param || :)
 vconsole.keymap=us rhgb quiet mem=1024M"
GRUB_DISABLE_RECOVERY="true"
```

See the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*, for more information about working with the **GRUB2** boot loader.

Part II. Advanced Installation Options

This part of the *Fedora Installation Guide* covers more advanced topics, which will be useful to users who need to perform a non-standard installation. It contains information about custom boot options, automating the installation using Kickstart, booting from a network location instead of local media, etc.

Boot Options

The **Anaconda** installer includes a range of boot options for administrators, which modify the default behavior of the installation program by enabling or disabling certain functions. To use one or more boot options, you either have to boot from installation media and append these options at the boot menu (see *Section 4.2, “The Boot Menu”*), or you must add them into your PXE server configuration file if you are booting from a network (see *Chapter 10, Setting Up an Installation Server*).

You can use multiple options at the same time; in that case, separate them by a single space.

There are two basic types of options described in this chapter:

- Options presented as ending with an "equals" sign (**=**) require a value to be specified - they cannot be used on their own. For example, the **`inst.vncpassword=`** option must also contain a value (in this case, a password). The correct form is therefore **`inst.vncpassword=`*`password`***. On its own, without a password specified, the option is invalid.

- Options presented without the "**=**" sign do not accept any values or parameters. For example, the **`rd.live.check`** option forces **Anaconda** to verify the installation media before starting the installation; if this option is present, the check will be performed, and if it is not present, the check will be skipped.

In addition to the options described in this chapter, the boot prompt also accepts **dracut** kernel options. A list of these options is available as the `dracut.cmdline(7)` man page.

Note

Boot options specific to the installation program always start with **`inst.`** in this guide. Currently, this prefix is optional - for example, **`resolution=1024x768`** will work exactly the same as **`inst.resolution=1024x768`**. However, it is expected that the **`inst.`** prefix will be mandatory in future releases.

8.1. Configuring the Installation System at the Boot Menu

The exact way to specify custom boot options is differs based on your system's architecture, firmware and the method you use to boot the installation. If you are booting from local media, you can specify options in the boot menu, before you begin the installation; if you are booting from a network using a PXE server, you must add boot options into the boot loader configuration file before you boot the installation system. For specific instructions, see *Section 4.2, “The Boot Menu”* if you are booting from local media, and *Chapter 10, Setting Up an Installation Server* if you are booting from a server.

8.2. Available Boot Options

The following options are available in Fedora:

8.2.1. Specifying the Installation Source

`inst.repo=`

Specifies the installation source - that is, a location where the installation program can find the images and packages it requires. For example:

```
inst.repo=cdrom
```

The source must be either:

- an installable tree, which is a directory structure containing the installation program's images, packages and repodata as well as a valid **.treeinfo** file
- a DVD (a physical disk present in the system's DVD drive)
- an ISO image of the full Fedora installation DVD, placed on a hard drive or a network location accessible from the installation system

This option allows for the configuration of different installation methods using different formats. The syntax is described in the table below.

Table 8.1. Installation Sources

Installation source	Option format
Any CD/DVD drive	**inst.repo=cdrom**
Specific CD/DVD drive	**inst.repo=cdrom:*device***
Hard Drive	**inst.repo=hd:*device*:*/path***
HTTP Server	**inst.repo=http://*host/path***
HTTPS Server	**inst.repo=https://*host/path***
FTP Server	**inst.repo=ftp://*username*:*password*@*host/path***
NFS Server	**inst.repo=nfs:[*options*:]*server*:*/path*** [1]

[1] This option uses NFS protocol version 3 by default. To use a different version, add **+nfsvers=*X*** to *options*.

Disk device names may be specified using the following formats:

- Kernel device name, for example /dev/sda1 or sdb2
- File system label, for example LABEL=Flash or LABEL=RHEL7
- File system UUID, for example UUID=8176c7bf-04ff-403a-a832-9557f94e61db

Non-alphanumeric characters must be represented as **\x*NN***, where *NN* is the hexadecimal representation of the character. For example, **\x20** is a white space (" ").

inst.stage2=

Specifies the location of the installation program runtime image to be loaded. The syntax is the same as in *Table 8.1, “Installation Sources”*. This option expects a path to a directory containing a valid **.treeinfo** file; the location of the runtime image will be read from this file if found. If a **.treeinfo** file is not available, **Anaconda** will try to load the image from **LiveOS/squashfs.img**.

> **Note**
>
> By default, this boot option is used on the installation media and set to a specific label (for example, `inst.stage2=hd:LABEL=F21\x20Server.x86_64`). If you modify the default label of the file system containing the runtime image, you must also edit this option to match the new label.

`inst.dd=`

If you need to perform a driver update during the installation, use the `inst.dd=` option. It can be used multiple times. The location of a driver RPM package can be specified using any of the formats described in *Table 8.1, “Installation Sources”*. With the exception of the `inst.dd=cdrom` option, the device name must always be specified. For example:

```
inst.dd=hd:/dev/sdb1:/driver.rpm
```

Using this option without any parameters (only as `inst.dd`) will prompt the installation program to ask you for a driver update disk with an interactive menu.

> **Important**
>
> You should never attempt to perform a driver update during the installation unless a missing our faulty driver is preventing you from completing the installation. Updating drivers which are not essential during the installation should always be performed after the system is installed.

`inst.askmethod`

Prevents the installer from trying to automatically configure an installation source.

Normally, when you first enter the **Installation Summary** screen, the installer will attempt to configure an installation source based on the type of media you used to boot. The full Fedora Server DVD will configure the source as local media, the netinst ISO image will configure the closest network mirror, etc. This process takes time; therefore, if you plan to use a different source than the default, this option will allow you to enter the **Installation Source** screen immediately without having to wait for the initial configuration to finish.

Using this option will leave the installation source configuration in an error state - you must manually specify a source in the **Installation Source** screen to proceed with the installation. Alternatively, you can use the `inst.repo=` option described above to configure a specific source directly in the boot menu.

8.2.2. Kickstart Boot Options

`inst.ks=`

Gives the location of a Kickstart file to be used to automate the installation. Locations can be specified using any of the formats valid for `inst.repo=`. See *Table 8.1, “Installation Sources”* for valid formats.

If you only specify a device and not a path, the installation program will look for the Kickstart file in **/ks.cfg** on the specified device. If you use this option without specifying a device, the installation program will use the following:

```
inst.ks=nfs:next-server:/filename
```

In the above example, *next-server* is the DHCP **next-server** option or the IP address of the DHCP server itself, and *filename* is the DHCP **filename** option, or **/kickstart/**. If the given file name ends with the **/** character, ***ip*-kickstart** is appended. For example:

Table 8.2. Default Kickstart File Location

DHCP server address	Client address	Kickstart file location
192.168.122.1	192.168.122.100	192.168.122.1:**/kickstart/192.168.122.100-kickstart**

inst.ks.sendmac

Adds headers to outgoing HTTP requests with the MAC addresses of all network interfaces. For example:

```
X-RHN-Provisioning-MAC-0: eth0 01:23:45:67:89:ab
```

This can be useful when using **inst.ks=http** to provision systems.

inst.ks.sendsn

Adds a header to outgoing HTTP requests. This header will contain the system's serial number, read from **/sys/class/dmi/id/product_serial**. The header has the following syntax:

```
X-System-Serial-Number: R8VA23D
```

8.2.3. Console, Environment and Display Options

console=

This kernel option specifies a device to be used as the primary console. For example, to use a console on the first serial port, use **console=ttyS0**. This option should be used along with the **inst.text** option.

You can use this option multiple times. In that case, the boot message will be displayed on all specified consoles, but only the last one will be used by the installation program afterwards. For example, if you specify **console=ttyS0 console=ttyS1**, the installation program will only use **ttyS1**.

noshell

Disables access to the root shell during the installation. This is useful with automated (Kickstart) installations - if you use this option, a user can watch the installation progress, but they cannot interfere with it by accessing the root shell by pressing **Ctrl**+**Alt**+**F2**.

inst.lang=

Sets the language to be used during the installation. Language codes are the same as the ones used in the **lang** Kickstart command as described in *Section A.4.2, "lang (optional) - Configure Language During Installation"*. On systems where the *system-config-language* package is installed, a list of valid values can also be find in **/usr/share/system-config-language/locale-list**.

If you perform the installation interactively, the language you specified using this option will be selected by default, but you will still be able to change it at the beginning of the graphical installation.

`inst.geoloc=`

Configures geolocation usage in the installation program. Geolocation is used to pre-set the language and time zone, and uses the following syntax: **`inst.geoloc=`*`value`***

The *`value`* parameter can be any of the following:

Table 8.3. Valid Values for the inst.geoloc Option

Disable geolocation	**`inst.geoloc=0`**
Use the Fedora GeoIP API	**`inst.geoloc=provider_fedora_geoip`**
Use the Hostip.info GeoIP API	**`inst.geoloc=provider_hostip`**

If this option is not specified, **Anaconda** will use **`provider_fedora_geoip`**.

`inst.keymap=`

Specifies the keyboard layout to be used by the installation program. Layout codes are the same as the ones used in the **`keyboard`** Kickstart command as described in *Appendix A, Kickstart Syntax Reference*.

`inst.text`

Forces the installation program to run in text mode instead of graphical mode. The text user interface is limited, for example, it does not allow you to modify the partition layout or set up LVM. When installing a system on a machine with a limited graphical capabilities, it is recommended to use VNC as described in *Section 8.2.6, “Enabling Remote Access Using VNC”*.

`inst.cmdline`

Forces the installation program to run in command line mode. This mode does not allow any interaction, all options must be specified in a Kickstart file or on the command line.

`inst.graphical`

Forces the installation program to run in graphical mode. This mode is the default.

`inst.resolution=`

Specifies the screen resolution in graphical mode. The format is *`N`*x*`M`*, where *`N`* is the screen width and *`M`* is the screen height (in pixels). The lowest supported resolution is **`640x480`**.

`inst.xdriver=`

Specifies the name of the X driver to be used both during the installation and on the installed system.

`inst.usefbx`

Tells the installation program to use the frame buffer X driver instead of a hardware-specific driver. This option is equivalent to **`inst.xdriver=fbdev`**.

`modprobe.blacklist=`

Blacklists (completely disables) one or more drivers. Drivers (mods) disabled using this option will be prevented from loading when the installation starts, and after the installation finishes, the installed system will keep these settings. The blacklisted drivers can then be found in the **`/etc/modprobe.d/`** directory.

Use a comma-separated list to disable multiple drivers. For example:

```
modprobe.blacklist=ahci,firewire_ohci
```

inst.sshd

Starts the `sshd` service during the installation, which allows you to connect to the system during the installation using SSH and monitor its progress. For more information on SSH, see the `ssh(1)` man page and the corresponding chapter in the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*.

> **Note**
>
> During the installation, the `root` account has no password by default. You can set a root password to be used during the installation with the **sshpw** Kickstart command as described in *Section A.6.6, “sshpw (optional) - Restrict ssh Access During Installation”*.

8.2.4. Network Boot Options

Initial network initialization is handled by **dracut**. This section only lists some of the more commonly used options; for a complete list, see the `dracut.cmdline(7)` man page. Additional information on networking is also available in the *Fedora Networking Guide*, available at *http://docs.fedoraproject.org/*.

ip=

Configures one or more network interfaces. To configure multiple interfaces, use the **ip** option multiple times - once for each interface. If multiple interfaces are configured, you must specify a primary boot interface using the **bootdev** option described below.

The following table lists valid values for this option:

Table 8.4. Network Interface Configuration Formats

Configuration Method	Option format
Automatic configuration of any interface	**ip=*method***
Automatic configuration of a specific interface	**ip=*interface*:*method***
Static configuration	**ip=*ip*::*gateway*:*netmask*:*hostname*:*interface*:none**
Automatic configuration of a specific interface with an override [1]	**ip=*ip*::*gateway*:*netmask*:*hostname*:*interface*:*method*:*mtu***

[1] Brings up the specified interface using the specified method of automatic configuration, such as **dhcp**, but overrides the automatically obtained IP address, gateway, netmask, hostname or other specified parameter. All parameters are optional; only specify the ones you wish to override and automatically obtained values will be used for the others.

The **method** parameter can be any the following:

Table 8.5. Automatic Interface Configuration Methods

Automatic configuration method	Value
DHCP	**dhcp**
IPv6 DHCP	**dhcp6**
IPv6 automatic configuration	**auto6**

Automatic configuration method	Value
iBFT (iSCSI Boot Firmware Table)	**ibft**

Note

If you use a boot option which requires network access, such as **inst.ks=http://*host*:/*path***, without specifying the **ip** option, the installation program will use **ip=dhcp**.

In the above tables, the *ip* parameter specifies the client's IP address. IPv6 addresses can be specified by putting them in square brackets, for example, **[2001:DB8::1]**.

The *gateway* parameter is the default gateway. IPv6 addresses are accepted here as well.

The *netmask* parameter is the netmask to be used. This can either be a full netmask (for example **255.255.255.0**) or a prefix (for example **64**).

The *hostname* parameter is the host name of the client system. This parameter is optional.

nameserver=
Specifies the address of the name server. This option can be used multiple times.

bootdev=
Specifies the boot interface. This option is mandatory if you use more than one **ip** option.

ifname=
Assigns a given interface name to a network device with a given MAC address. Can be used multiple times. The syntax is **ifname=*interface*:*MAC***. For example:

```
ifname=eth0:01:23:45:67:89:ab
```

inst.dhcpclass=
Specifies the DHCP vendor class identifier. The dhcpd service will see this value as **vendor-class-identifier**. The default value is **anaconda-$(uname -srm)**.

vlan=
Sets up a Virtual LAN (VLAN) device on a specified interface with a given name. The syntax is **vlan=*name*:*interface***. For example:

```
vlan=vlan5:em1
```

The above will set up a VLAN device named **vlan5** on the **em1** interface. The *name* can take the following forms:

Table 8.6. VLAN Device Naming Conventions

Naming scheme	Example
VLAN_PLUS_VID	**vlan0005**
VLAN_PLUS_VID_NO_PAD	**vlan5**
DEV_PLUS_VID	**em1.0005**.

Naming scheme	Example
DEV_PLUS_VID_NO_PAD	**`em1.5`**.

`bond=`

Set up a bonding device with the following syntax: **`bond=`*`name`*`[:`*`slaves`*`][:`*`options`*`]`**. Replace *`name`* with the bonding device name, *`slaves`* with a comma-separated list of physical (ethernet) interfaces, and *`options`* with a comma-separated list of bonding options. For example:

```
bond=bond0:em1,em2:mode=active-backup,tx_queues=32,downdelay=5000
```

For a list of available options, execute the **`modinfo bonding`** command.

Using this option without any parameters will assume **`bond=bond0:eth0,eth1:mode=balance-rr`**.

`team=`

Set up a team device with the following syntax: **`team=`*`master`*`:`*`slaves`***. Replace *`master`* with the name of the master team device and *`slaves`* with a comma-separated list of physical (ethernet) devices to be used as slaves in the team device. For example:

```
team=team0:em1,em2
```

8.2.5. Advanced Installation Options

`inst.multilib`

Configure the system for multilib packages (that is, to allow installing 32-bit packages on a 64-bit x86 system) and install packages specified in this section as such.

Normally, on an AMD64 or Intel 64 system, only packages for this architecture (marked as **`x86_64`**) and packages for all architectures (marked as **`noarch`** would be installed. When you use this option, packages for 32-bit AMD or Intel systems (marked as **`i686`**) will be automatically installed as well if available.

This only applies to packages directly specified in the **`%packages`** section. If a package is only installed as a dependency, only the exact specified dependency will be installed. For example, if you are installing package *foo* which depends on package *bar*, the former will be installed in multiple variants, while the latter will only be installed in variants specifically required.

`inst.gpt`

Force the installation program to install partition information into a GUID Partition Table (GPT) instead of a Master Boot Record (MBR). This option is meaningless on UEFI-based systems, unless they are in BIOS compatibility mode.

Normally, BIOS-based systems and UEFI-based systems in BIOS compatibility mode will attempt to use the MBR schema for storing partitioning information, unless the disk is larger than 2 TB. Using this option will change this behavior, allowing a GPT to be written even to disks smaller than 2 TB.

See *Section 5.4.10.6, “Recommended Partitioning Scheme”* for more information about GPT and MBR, and *Section B.1.4, “GUID Partition Table (GPT)”* for more general information about GPT, MBR and disk partitioning in general.

`inst.zram`

This option controls the usage of *zRAM* swap during the installation. It creates a compressed block device inside the system RAM and uses it for swap space instead of the hard drive. This allows the installer to essentially increase the amount of memory available, which makes the installation faster on systems with low memory.

By default, swap on zRAM is enabled on systems with 2 GB or less RAM, and disabled on systems with more than 2 GB of memory. You can use this option to change this behavior - on a system with more than 2 GB RAM, use **`inst.zram=1`** to enable it, and on systems with 2 GB or less memory, use **`inst.zram=0`** to disable this feature.

`inst.memcheck`

Perform a check at the beginning of the installation to determine if there is enough available RAM. If there is not enough memory detected, the installation will stop with an error message. This option is enabled by default; use **`inst.memcheck=0`** to disable it.

`inst.nodnf`

Use the older **Yum** backend for package installation and dependency resolution instead of **DNF**. See the *Fedora System Administrator's Guide*, available at https://docs.fedoraproject.org/, for details about **DNF**.

`inst.kdump_addon=`

Turns the **Kdump** installer add-on **on** or **`off`** (the add-on is off by default). If you use **`inst.kdump_addon=on`**, a new screen, which allows you to configure the **Kdump** kernel crash dumping mechanism, will appear in the main menu of the graphical or text interface (see *Section 5.4.11, “Kdump”*). Enabling this add-on is also necessary to use the *Section A.7.1, “%addon com_redhat_kdump (optional) - Configure kdump”* Kickstart command.

8.2.6. Enabling Remote Access Using VNC

The following options are necessary to configure **Anaconda** for remote graphical installation. See *Chapter 11, Installing Using VNC* for more details.

`inst.vnc`

Specifies that the installation program's graphical interface should be run in a `VNC` session. If you specify this option, you will need to connect to the system using a `VNC` client application to be able to interact with the installation program. `VNC` sharing is enabled, so multiple clients can connect to the system at the same time.

Note

A system installed using `VNC` will start in text mode by default.

`inst.vncpassword=`

`VNC` server used by the installation program. Any `VNC` client attempting to connect to the system will have to provide the correct password to gain access. For example, **`inst.vncpassword=`*`testpwd`*** will set the password to **`testpwd`**. The password must be between 6 and 8 characters long.

Note

If you specify an invalid password (one that is too short or too long), you will be prompted to specify a new one by a message from the installation program:

```
VNC password must be six to eight characters long.
Please enter a new one, or leave blank for no password.

Password:
```

`inst.vncconnect=`

Connect to a listening VNC client at a specified host and port once the installation starts. The correct syntax is **`inst.vncconnect=`*`host`*`:`*`port`***. The *`port`* parameter is optional - if you do not specify one, the installation program will use **`5900`**.

8.2.7. Debugging and Troubleshooting

`inst.updates=`

Specifies the location of the **`updates.img`** file to be applied to the installation program runtime. The syntax is the same as in the **`inst.repo`** option - see *Table 8.1, "Installation Sources"* for details. In all formats, if you do not specify a file name but only a directory, the installation program will look for a file named **`updates.img`**.

`inst.loglevel=`

Specifies the minimum level for messages to be logged on a terminal. This only concerns real-time logging in a terminal; log files will always contain messages of all levels.

Possible values for this option from the lowest to highest level are: **`debug`**, **`info`**, **`warning`**, **`error`** and **`critical`**. The default value is **`info`**, which means that by default, the logging terminal will display messages ranging from **`info`** to **`critical`**, but not **`debug`**.

`inst.syslog=`

Once the installation starts, this options sends log messages to the syslog process on the specified host. The remote syslog process must be configured to accept incoming connections.

`inst.virtiolog=`

Specifies a `virtio` port (a character device at **`/dev/virtio-ports/`*`name`***) to be used for forwarding logs. The default value is **`org.fedoraproject.anaconda.log.0`**; if this port is present, it will be used.

8.3. Deprecated Boot Options

Options in this list are *deprecated*. They will still work, but there are other options which offer the same functionality and should be preferred. Using deprecated options is not recommended and they are expected to be removed in future releases.

Note

Note that as described in the introduction to this chapter, options specific to the installation program now use the **`inst.`** prefix. For example, the **`vnc=`** option is considered deprecated and replaced by the **`inst.vnc=`** option. However, these changes are not listed here.

`inst.headless`

Used to specify that the machine being installed onto does not have any display hardware, and to prevent the installer from trying to detect a display.

`method=`

Configured the installation method. Use the **`inst.repo=`** option instead.

`repo=nfsiso:`*`server`*`:/`*`path`*

In NFS installations, specified that the target is an ISO image located on an NFS server instead of an installable tree. The difference is now detected automatically, which means this option is the same as **`inst.repo=nfs:`*`server`*`:/`*`path`***.

`dns=`

Configured the Domain Name Server (DNS). Use the **`nameserver=`** option instead.

`netmask=`, **`gateway=`**, **`hostname=`**, **`ip=`**, **`ipv6=`**

These options have been consolidated under the **`ip=`** option.

`ksdevice=`

Select network device to be used at early stage of installation. Different values have been replaced with different options; see the table below.

Table 8.7. Automatic Interface Configuration Methods

Value	Current behavior
Not present	All devices are attempted to be activated using `dhcp`, unless desired device and configuration is specified by **`ip=`** option and/or the **`BOOTIF`** option.
`ksdevice=link`	Similar to the above, with the difference that network will always be activated in the initramfs, whether it is needed or not. The supported **`rd.neednet`** option (provided by **dracut**) should be used instead.
`ksdevice=bootif`	Ignored (the BOOTID= option is used by default when specified)
`ksdevice=ibft`	Replaced with the **`ip=ibft`** option
`ksdevice=`*`MAC`*	Replaced with **`BOOTIF=`*`MAC`***
`ksdevice=`*`device`*	Replaced by specifying the device name using the **`ip=`** option.

> **Important**
>
> When performing a Kickstart installation, booting from local media and having the Kickstart file on local media as well, the network will not be initialized. This means that any other Kickstart options requiring network access, such as pre-installation or post-installation scripts accessing a network location, will cause the installation to fail. This is a known issue; see BZ#*1085310*[1] for details.
>
> To work around this issue, either use the **`ksdevice=link`** boot option, or add the **`--device=link`** option to the **`network`** command in your Kickstart file.

`blacklist=`
: Used to disable specified drivers. This is now handled by the **`modprobe.blacklist=`** option.

`nofirewire=`
: Disabled support for the FireWire interface. You can disable the FireWire driver (**`firewire_ohci`**) by using the **`modprobe.blacklist=`** option instead:

```
modprobe.blacklist=firewire_ohci
```

8.4. Removed Boot Options

The following options are removed. They were present in previous releases of Fedora, but they cannot be used anymore.

`asknetwork`
: The installation program's `initramfs` is now completely non-interactive, which means that this option are not available anymore. Use **`ip=`** to configure network settings.

`serial`
: This option forced **Anaconda** to use the **`/dev/ttyS0`** console as the output. Use the **`console=/dev/ttyS0`** (or similar) instead.

`updates=`
: Specified the location of updates for the installation program. Use the **`inst.updates=`** option instead.

`essid=`, **`wepkey=`**, **`wpakey=`**
: Configured wireless network access. Network configuration is now being handled by **dracut**, which does not support wireless networking, rendering these options useless.

`ethtool=`
: Used in the past to configure additional low-level network settings. All network settings are now handled by the **`ip=`** option.

`gdb`
: Allowed you to debug the loader. Use **`rd.debug`** instead.

[1] https://bugzilla.redhat.com/show_bug.cgi?id=1085310

`mediacheck`

Verified the installation media before starting the installation. Replaced with the **`rd.live.check`** option.

`ks=floppy`

Specified a floppy disk as the Kickstart file source. Floppy drives are not supported anymore.

`display=`

Configured a remote display. Replaced with the **`inst.vnc`** option.

`utf8`

Added UTF8 support when installing in text mode. UTF8 support now works automatically.

`noipv6`

Used to disable IPv6 support in the installation program. IPv6 is now built into the kernel so the driver cannot be blacklisted; however, it is possible to disable IPv6 using the **`ipv6.disable`** option.

`upgradeany`

Upgrades are done in a different way in current Fedora releases. For more information about upgrading your system, see *Chapter 12, Upgrading Your Current System*.

`vlanid=`

Used to configure Virtual LAN (802.1q tag) devices. Use the **`vlan=`** option instead.

`inst.dnf`

Used to enable **DNF** to handle package installation and dependency resolution in Fedora 21. **DNF** is now used by default and this option has been replaced with **`inst.nodnf`**, which reverts back to **Yum**.

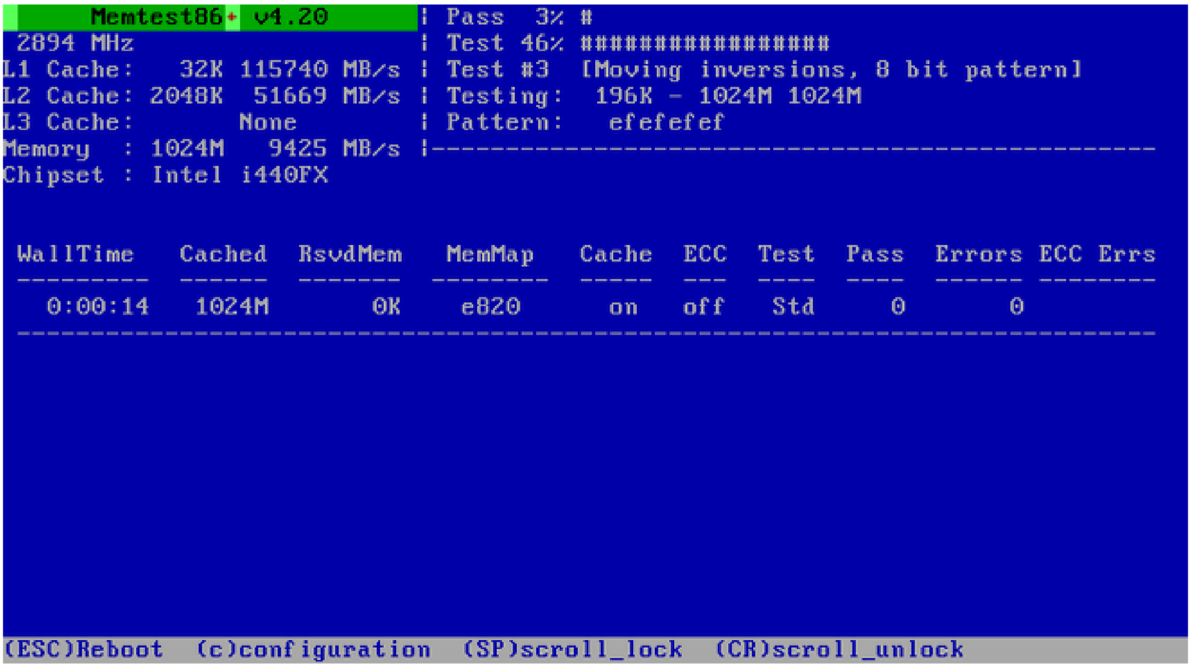
8.5. Using the Maintenance Boot Modes

8.5.1. Loading the Memory (RAM) Testing Mode

Faults in memory (RAM) modules may cause your system to freeze or crash unpredictably. In some cases, memory faults may only cause errors with particular combinations of software. For this reason, you should test the memory of a computer before you install Fedora for the first time, even if it has previously run other operating systems.

Fedora includes the **Memtest86+** memory testing application. To start memory testing mode, choose **Troubleshooting** > **Memory test** at the boot menu. Testing will begin immediately. By default, **Memtest86+** carries out ten tests in every pass; a different configuration can be specified by accessing the configuration screen using the **c** key. After the first pass completes, a message will appear at the bottom informing you of the current status, and another pass will start automatically.

> **Note**
>
> **Memtest86+** only works on systems with BIOS firmware. Support for UEFI systems is currently unavailable.



```
      Memtest86+ v4.20      | Pass  3% #
 2894 MHz                   | Test 46% #################
L1 Cache:   32K 115740 MB/s | Test #3  [Moving inversions, 8 bit pattern]
L2 Cache: 2048K  51669 MB/s | Testing:  196K - 1024M 1024M
L3 Cache:       None        | Pattern:   efefefef
Memory  : 1024M   9425 MB/s |-------------------------------------------------
Chipset : Intel i440FX

 WallTime   Cached  RsvdMem   MemMap   Cache  ECC  Test  Pass  Errors ECC Errs
 ---------  ------  -------  --------  -----  ---  ----  ----  ------ --------
   0:00:14   1024M       0K    e820      on   off   Std     0       0
 -----------------------------------------------------------------------------

(ESC)Reboot  (c)configuration  (SP)scroll_lock  (CR)scroll_unlock
```

Figure 8.1. Memory Check Using Memtest86+

The main screen displayed while testing is in progress is divided into three main areas:

- The upper left corner shows information about your system's memory configuration - the amount of detected memory and processor cache and their throughputs and processor and chipset information. This information is detected when **Memtest86+** starts.

- The upper right corner displays information about the tests - progress of the current pass and the currently running test in that pass as well as a description of the test.

- The central part of the screen is used to display information about the entire set of tests from the moment when the tool has started, such as the total time, the number of completed passes, number of detected errors and your test selection. On some systems, detailed information about the installed memory (such as the number of installed modules, their manufacturer, frequency and latency) will be also displayed here. After the each pass completes, a short summary will appear in this location. For example:

  ```
  ** Pass complete, no errors, press Esc to exit **
  ```

  If **Memtest86+** detects an error, it will also be displayed in this area and highlighted red. The message will include detailed information such as which test detected a problem, the memory location which is failing, and others.

In most cases, a single successful pass (that is, a single run of all 10 tests) is sufficient to verify that your RAM is in good condition. In some rare circumstances, however, errors that went undetected on the first pass might appear on subsequent passes. To perform a thorough test on an important system, leave the tests running overnight or even for a few days in order to complete multiple passes.

Note

The amount of time it takes to complete a single full pass of **Memtest86+** varies depending on your system's configuration (notably the RAM size and speed). For example, on a system with 2 GB of DDR2 memory at 667 MHz, a single pass will take roughly 20 minutes to complete.

To halt the tests and reboot your computer, press the **`Esc`** key at any time.

For more information about using **Memtest86+**, see the official website at *http://www.memtest.org/*. A **`README`** file is also located in **`/usr/share/doc/memtest86+-`*`version`*`/`** on Fedora systems with the *memtest86+* package installed.

8.5.2. Verifying Boot Media

You can test the integrity of an ISO-based installation source before using it to install Fedora. These sources include DVDs and ISO images stored on a local hard drive or NFS server. Verifying that the ISO images are intact before you attempt an installation helps to avoid problems that are often encountered during installation.

To test the integrity of an ISO image, append the **`rd.live.check`** to the boot loader command line. Note that this option is used automatically if you select the default installation option from the boot menu (**Test this media & install Fedora**).

8.5.3. Booting Your Computer in Rescue Mode

You may boot a command-line Linux system from an installation disc without actually installing Fedora on the computer. This enables you to use the utilities and functions of a running Linux system to modify or repair already installed operating systems.

To load the rescue system with the installation disk or USB drive, choose **Rescue a Fedora system** from the **Troubleshooting** submenu in the boot menu, or use the **`inst.rescue`** boot option.

Specify the language, keyboard layout and network settings for the rescue system with the screens that follow. The final setup screen configures access to the existing system on your computer.

By default, rescue mode attaches an existing operating system to the rescue system under the directory **`/mnt/sysimage/`**.

Automating the Installation with Kickstart

Kickstart installations offer a means to automate the installation process, either partially or fully. Kickstart files contain answers to all questions normally asked by the installation program, such as what time zone do you want the system to use, how should the drives be partitioned or which packages should be installed. Providing a prepared Kickstart file when the installation begins therefore allows the you to perform the installation automatically, without need for any intervention from the user. This is especially useful when deploying Fedora on a large number of systems at once.

All Kickstart scripts and the log files of their execution are stored in the **`/tmp`** directory to assist with debugging installation issues.

9.1. How to Perform a Kickstart Installation

Kickstart installations can be performed using a local DVD, a local hard drive, or via NFS, FTP, HTTP, or HTTPS.

To use Kickstart, you must:

1. Create a Kickstart file.

2. Create boot media or configure a network boot (PXE) server which will be used to begin the installation.

3. Make the Kickstart file available on removable media, a hard drive, or a network location.

4. Start the Kickstart installation by booting the installer and using a boot option to tell the installer where to find the Kickstart file.

This chapter explains these steps in detail.

9.1.1. Creating a Kickstart File

The Kickstart file itself is a plain text file, containing keywords listed in *Appendix A, Kickstart Syntax Reference*, which serve as directions for the installation. Any text editor able to save files as ASCII text (such as **Gedit** or **vim** on Linux systems or **Notepad** on Windows systems) can be used to create and edit Kickstart files.

The recommended approach to creating Kickstart files is to perform a manual installation on one system first. After the installation completes, all choices made during the installation are saved into a file named **`anaconda-ks.cfg`**, located in the **`/root/`** directory on the installed system. You can then copy this file, make any changes you need, and use the resulting configuration file in further installations.

When creating a Kickstart file, keep in mind the following:

- Lines starting with a pound sign (#) are treated as comments and are ignored.

- Sections must be specified *in order*. Items within the sections do not have to be in a specific order unless otherwise specified. The correct section order is:

  - The command section which contains actual Kickstart commands and options as listed in *Appendix A, Kickstart Syntax Reference*. Note that some commands, such as **`install`**, are mandatory, but most commands are optional.

  - The **`%packages`** section which contains a list of packages and package groups to be installed. See *Section A.12, “%packages (required) - Package Selection”* for details.
  - The **`%pre`** and **`%post`** sections, containing a pre-installation and post-installation scripts. These two sections can be in any order and are not mandatory. See *Section A.13, “%pre (optional) - Pre-installation Script”* and *Section A.14, “%post (optional) - Post-installation Script”* for details.

  Important

  The **`%packages`**, **`%pre`** and **`%post`** sections must end with **`%end`**, otherwise the installation program will refuse the Kickstart file. The main command section has no special ending statement.

- Omitting any required item results in the installation program prompting the user for an answer to the related item, just as the user would be prompted during a typical installation. Once the answer is given, the installation will continue. Note that if the system you are installing has no display, you will not be able to see the prompt, and the installation will appear to have failed.

9.1.2. Verifying the Kickstart File

When creating or customizing your kickstart file, it is useful to verify that it is valid before attempting to use it in an installation. Fedora includes the **ksvalidator** command line utility which can be used to do this. This tool is a part of the *pykickstart* package. To install this package, execute the following command:

```
# dnf install pykickstart
```

After installing the package, you can validate a Kickstart file using the following command:

```
$ ksvalidator /path/to/kickstart.ks
```

Replace *`/path/to/kickstart.ks`* with the path to the Kickstart file you want to verify.

For more information about this tool, see the `ksvalidator(1)` man page.

Important

Keep in mind that the validation tool has its limitations. The Kickstart file can be very complicated; **ksvalidator** can make sure the syntax is correct and that the file does not include removed options, but it cannot guarantee the installation will be successful. It also does not attempt to validate the **`%pre`**, **`%post`** and **`%packages`** sections of the Kickstart file.

9.1.3. Making the Kickstart File Available

Once you create a Kickstart file, you can place it in one of the following locations:

- On *removable media*, such as a DVD or USB flash drive connected to the installation system

- On a *hard drive* connected to the installation system
- On a *network share* reachable from the installation system

Normally, a Kickstart file is copied to removable media or a hard drive, or made available on the network. Placing the file in a network location complements the usual approach to Kickstart installations, which is also network-based: the system is booted using a PXE server, the Kickstart file is downloaded from a network share, and software packages specified in the file are downloaded from remote repositories.

Chapter 10, Setting Up an Installation Server offers some additional information about preparing for a network-based installation.

9.1.4. Starting the Kickstart Installation

Once you have everything ready - you have created a valid Kickstart file and you have either local boot media or a PXE server available, you can start the Kickstart installation. You need to use the **`inst.ks=`** boot option either in the boot menu (when booting from local media), or add this option to your PXE server configuration. For information about boot options used in Kickstart installations, see *Section 8.2.2, “Kickstart Boot Options”*.

Setting Up an Installation Server

Note

This appendix is intended for users with previous Linux experience. If you are a new user, you may want to install using minimal boot media or the distribution DVD instead.

10.1. PXE Installation Overview

Preboot Execution Environment, or PXE, is a techonology that allows computers to boot directly from resources provided over the network. Installing Fedora over the network means you don't have to create media, and you can install to multiple computers or virtual machine simultaneously. The process involves a number of components and features working together to provide the resources required.

PXE-capable computer

Most modern computers have the capability to network boot. Typically, a function key pressed during boot will bring up a boot selection menu. In environments designed for unattended administration, systems will often be configured to first attempt booting from the network, then boot from local storage, and the installation server is configured to only offer the installation when required. Your computer's manual will provide specific instructions on setting boot priorities.

DHCP Server

When a system requests an address during network booting, the DHCP server also provides the location of files to boot. A network should have only one DHCP server.

TFTP Server

Because the pre-boot environment is very simple, files must be provided in a very simple way. Trivial File Transfer Protocol, or TFTP, provides the system with the bootloader required to continue the installation process.

Bootloader

Because the job of booting an operating system is too complex for the pre-boot environment, a bootloader is used to load the kernel and related files. It also provides configuration information to the installer, and can offer a menu to select from different configurations.

Kernel and Initramfs

The kernel is the core of any Linux operating system, and the initramfs provides the kernel with required tools and resources. These files are also provided by tftp.

Package repository

A Fedora repository must be available for the installation. The example in this section uses the public Fedora mirrors as the repository source, but you can also use a repo on the local network provided by NFS, FTP, or HTTP. Repositories can be configured using the **`inst.repo=`** boot option; see *Section 8.2.1, “Specifying the Installation Source”* for details.

10.2. DHCP Server Configuration

Procedure 10.1. Installing and configuring dhcpd

1. Install the dhcp server package.

   ```
   # dnf install dhcp
   ```

2. Create a simple configuration for the dhcp server at `/etc/dhcp/dhcpd.conf`

   ```
   subnet 192.168.1.0 netmask 255.255.255.0 {
   authoritative;
   default-lease-time 600;
   max-lease-time 7200;
   ddns-update-style none;

   option domain-name-servers 192.168.1.1;
   option routers 192.168.1.1;

   }
   ```

3. Test your configuration and address any problems you discover.

   ```
   systemctl start dhcpd
   journalctl --unit dhcpd --since -2m --follow
   ```

4. Add entries to point clients to their bootloader and the server that provides it to your subnet configuration in `/etc/dhcp/dhcpd.conf`. Because DHCP clients provide the server with identifying information along with their address request, BIOS clients and UEFI clients can each be directed to the correct bootloader.

   ```
   # refer to RFC4758 for possible arch option values
   option arch code 93 = unsigned integer 16;

   subnet 192.168.1.0 netmask 255.255.255.0 {
   if option arch = 00:07 {
   filename "uefi/shim.efi";
   } else {
   filename "pxelinux.0";
   }

   next-server 192.168.1.2;

   ...
   ```

5. Restart the dhcp service to check the configuration and make changes as needed.

   ```
   systemctl restart dhcpd
   journalctl --unit dhcpd --since -2m --follow
   ```

10.3. Installing the tftp server

Procedure 10.2. Installing the tftp server

1. Install the tftp server package.

```
# dnf install tftp-server
```

2. Start and enable the `tftp socket`. `systemd` will automatically start the `tftpd` service when required.

```

systemctl start tftp.socket
systemctl enable tftp.socket
```

10.4. Providing and configuring bootloaders for PXE clients

Procedure 10.3. Getting the bootloader files

1. Get the *syslinux* bootloader for BIOS clients.

   a. Install the *syslinux* package.

   ```
   # dnf install syslinux
   ```

   b. Create a directory for the bootloader files, and make them available there.

   ```

   mkdir -p /var/lib/tftpboot/pxelinux.cfg
   cp /usr/share/syslinux/
   {pxelinux.0,vesamenu.c32,ldlinux.c32,libcom32.c32,libutil.c32} /var/lib/tftpboot/
   ```

2. Get the bootloader files for UEFI systems

   a. Install the *shim* and *grub2-efi* packages. If your server is a BIOS system, you *must install the packages to a temporary install root*. Installing them directly on a BIOS machine will attempt to configure the system for UEFI booting and cause problems.

   ```
   # dnf install shim grub2-efi --installroot=/tmp/fedora --releasever 22
   ```

   b. Create a directory for the bootloader files, and make them available there.

   ```

   mkdir -p /var/lib/tftpboot/uefi
   cp /tmp/fedora/boot/efi/EFI/fedora/{shim.efi,grubx64.efi} /var/lib/tftpboot/uefi/
   ```

Procedure 10.4. Configuring client bootloaders

1. Create a boot menu for BIOS clients at **`/var/lib/tftpboot/pxelinux.cfg/default`**.

```

default vesamenu.c32
prompt 1
timeout 600

label linux
```

```
menu label ^Install Fedora 22 64-bit
menu default
kernel f22/vmlinuz
append initrd=f22/initrd.img inst.stage2=http://download.fedoraproject.org/pub/fedora/
linux/releases/22/Server/x86_64/os/ ip=dhcp

label server
menu label ^Install Fedora 22 ( Minimal Image )
menu default
kernel f22/vmlinuz
append initrd=f22/initrd.img inst.stage2=http://download.fedoraproject.org/pub/fedora/
linux/releases/22/Server/x86_64/os/ ip=dhcp ks=https://example.com/fedora/kickstarts/
minimal.ks

label rescue
menu label ^Rescue installed system
kernel f22/vmlinuz
append initrd=f22initrd.img ip=dhcp root=live:http://download.fedoraproject.org/pub/
fedora/linux/releases/22/Server/x86_64/os/LiveOS/squashfs.img rescue

label local
menu label Boot from ^local drive
localboot 0xffff
```

2. Create a boot menu for UEFI clients at **`/var/lib/tftpboot/pxelinux/uefi`**.

```

function load_video {
insmod efi_gop
insmod efi_uga
insmod video_bochs
insmod video_cirrus
insmod all_video
}

load_video
set gfxpayload=keep
insmod gzio

menuentry 'Install Fedora 64-bit'  --class fedora --class gnu-linux --class gnu --class
 os {
linuxefi f22/vmlinuz ip=dhcp inst.repo=http://download.fedoraproject.org/pub/fedora/
linux/releases/22/Server/x86_64/os/
initrdefi f22/initrd.img
}

menuentry 'Install Fedora 22 Server'  --class fedora --class gnu-linux --class gnu --
class os {
kernel f22/vmlinuz
append initrd=f22/initrd.img inst.repo=http://download.fedoraproject.org/pub/fedora/
linux/releases/22/Server/x86_64/os/ ip=dhcp ks=https://git.fedorahosted.org/cgit/spin-
kickstarts.git/plain/fedora-install-server.ks?h=f21
}

menuentry 'Rescue installed system'  --class fedora --class gnu-linux --class gnu --
class os {
kernel f22/vmlinuz
append f22/initrd=initrd.img root=live:http://download.fedoraproject.org/pub/fedora/
linux/releases/22/Server/x86_64/os/LiveOS/squashfs.img rescue
}
```

10.5. Getting the kernel and initrd

Procedure 10.5. Downloading the kernel and initrd

1. Create a directory for the files.

   ```
   mkdir -p /var/lib/tftpboot/f22
   ```

2. Download the kernel.

   ```
   wget http://download.fedoraproject.org/pub/fedora/linux/releases/22/Server/x86_64/os/
   images/pxeboot/vmlinuz -O /var/lib/tftpboot/f22/vmlinuz
   ```

3. Download the initrd

   ```
   wget http://download.fedoraproject.org/pub/fedora/linux/releases/22/Server/x86_64/os/
   images/pxeboot/initrd.img -O /var/lib/tftpboot/f22/initrd.img
   ```

10.6. Providing repositories

The examples in this section use the public Fedora mirrors as the package source. For faster installations, installing to many systems, or more isolated environments, you may wish to maintain a local repository.

Fedora Infrastructure maintains instructions for a configuring a local mirror at *https://fedoraproject.org/wiki/Infrastructure/Mirroring*. The preferred method for providing repositories is via HTTP, and you can refer to the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*, to configure `httpd`.

10.7. Advanced network installations with Cobbler

For more complex environments, Fedora offers the *cobbler* installation server. Tasks like managing kickstart configurtations, coordinating repositories, maintaining dns records, dhcp servers, and even puppet manifests are effectively automated by *cobbler*.

While levaraging all of the features provided by cobbler can be relatively simple, the full functionality of this powerful tool is too broad to be documented in this guide. The cobbler community provides excellent documentation at *http://www.cobblerd.org/manuals/2.6.0/* to accompany the packages in the Fedora repository.

Installing Using VNC

The graphical installation interface is the recommended method of installing Fedora. However, in some cases, accessing the graphical interface directly is difficult or impossible. Some systems lack the capability to connect a display and a keyboard, making VNC a necessity for manual (non-Kickstart) installations.

To allow manual installations on *headless systems* (systems without a directly connected display, keyboard and mouse), the **Anaconda** installation program includes a *Virtual Network Computing* (VNC) mode which allows the graphical mode of the installation program to run locally, but display on another system connected to the network. The VNC installation provides you with the full range of installation options.

This chapter provides instructions on activating VNC mode on the installation system and connecting to it using a VNC viewer.

11.1. Installing a VNC Viewer

Performing a VNC installation requires a VNC viewer running on your workstation or another terminal computer. VNC viewers are available in the repositories of most Linux distributions; free VNC viewers are also available for other operating systems such as Windows. On Linux systems, use your package manager to search for a viewer for your distribution.

The following VNC viewers are available in Fedora:

- **TigerVNC** - A basic viewer independent of your desktop environment. Installed as the *tigervnc* package.
- **Vinagre** - A viewer for the **GNOME** desktop environment. Installed as the *vinagre* package.
- **KRDC** - A viewer integrated with the **KDE** desktop environment. Installed as the *kdenetwork-krdc* package.

To install any of the viewers listed above, execute the following command as `root`:

```
# dnf install package
```

Replace *`package`* with the package name of the viewer you want to use (for example, *tigervnc*).

Note

Procedures in this chapter assume you are using **TigerVNC** as your VNC viewer. Specific instructions for other viewers may differ, but the general principles still apply.

11.2. Performing a VNC Installation

The **Anaconda** installation program offers two modes for VNC installation: *Direct mode* and *Connect mode*. The modes differ in the way the connection between the server and viewer is established. After you successfully connect, the installation will progress the same way regardless of the mode you used.

Direct Mode

In this mode, **Anaconda** is configured to start the installation and wait for an incoming connection from VNC viewer before proceeding. While waiting for an incoming connection, the system's IP address and the port on which the installer expects the connection is displayed on the display or console if available; this implies that you need at least a serial console to connect using this mode, but you can work around this limitation if you know the default VNC port and the system's IP address.

Connect Mode

In this mode, the VNC viewer is started on the remote system in *listening mode*. The VNC viewer waits for an incoming connection on a specified port. Then, **Anaconda** is started and the host name/IP address and port number of the viewer are provided using a boot option or a Kickstart command. When the installation begins, the installation program establishes a connection with the listening VNC viewer using the specified host name/IP address and port number. Connect mode is therefore easier to use on systems with no local display or console, but it also may require additional preparation, because the viewer system must be able to accept incoming connections on the specified port, which usually requires changing firewall settings.

11.2.1. Choosing a VNC Installation Mode

- Visual and Interactive access to the system
  - If visual and interactive access to the system being installed is not available, then you should use Connect Mode.
- Network Connection Rules and Firewalls
  - If the system being installed is not allowed inbound connections by a firewall, then you must use Connect Mode or disable the firewall. Disabling a firewall may have security implications.
  - If the remote system running the VNC viewer is not allowed incoming connections by a firewall, then you must use Direct Mode, or disable the firewall. Disabling a firewall may have security implications.

11.2.2. Installing in VNC Direct Mode

VNC Direct Mode is when the VNC viewer initiates a connection to the system being installed. **Anaconda** will tell you when to initiate this connection.

Procedure 11.1. Starting VNC in Direct Mode

1. Open the VNC viewer (for example, **TigerVNC**) on the workstation you will be using to connect to the system being installed. A window similar to *Figure 11.1, "TigerVNC Connection Details"* will be displayed with an input field allowing you to specify an IP address.

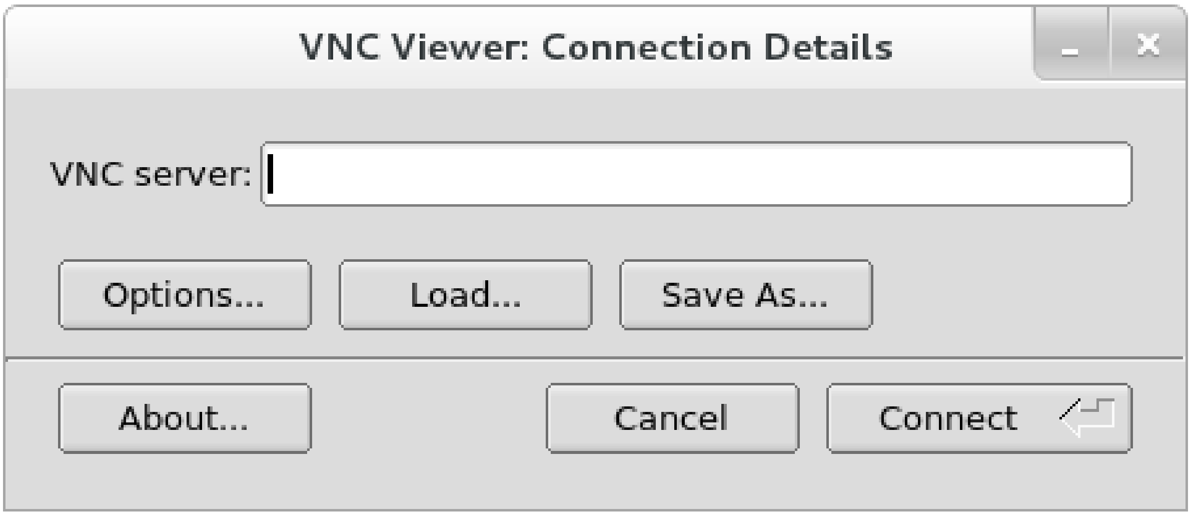


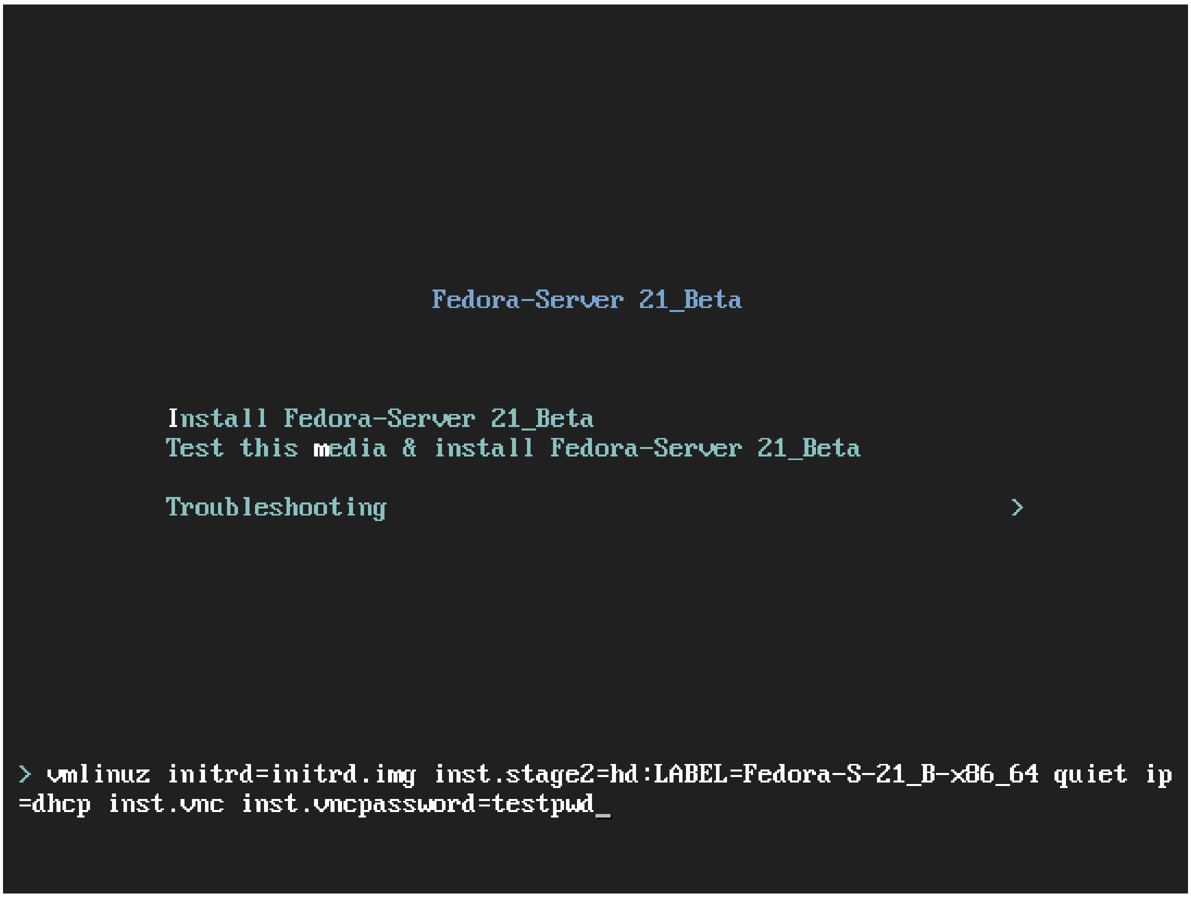
Figure 11.1. TigerVNC Connection Details

2. Boot the installation system and wait for the boot menu to appear. In the menu, edit boot options (see *Section 4.2, "The Boot Menu"*) and append the **`inst.vnc`** option to the end of the command line.

   Optionally, if you want to restrict VNC access to the installation system, add the **`inst.vncpassword=`*`PASSWORD`*** boot option as well. Replace *`PASSWORD`* with the password you want to use for the installation. The VNC password must be between 6 and 8 characters long.

   Important

   Use a temporary password for the **`inst.vncpassword=`** option. It should not be a real or root password you use on any system.


Fedora-Server 21_Beta
Install Fedora-Server 21_Beta
Test this media & install Fedora-Server 21_Beta
Troubleshooting
>

```
> vmlinuz initrd=initrd.img inst.stage2=hd:LABEL=Fedora-S-21_B-x86_64 quiet ip
=dhcp inst.vnc inst.vncpassword=testpwd_
```

Figure 11.2. Adding VNC Boot Options

3. Start the installation using the edited options. The system will initialize the installation program and start the necessary services. When the system is ready, you will see a message on the screen similar to the following:

   ```
   13:14:47 Please manually connect your VNC viewer to 192.168.100.131:5901 to begin the
    install.
   ```

   Note the IP address and port number (in the above example, `192.168.100.131:5901`).

4. On the system running the VNC Viewer, enter the IP address and port number obtained in the previous step into the **Connection Details** dialog in the same format as it was displayed on the screen by the installer. Then, click **Connect**. The VNC viewer will now connect to the installation system. If you set up a VNC password, enter it when prompted and press **OK**.

When the connection is successfully established, a new window will open on the system running the VNC viewer, displaying the installation menu. This window will provide full remote access to the installer until the installation finishes and the system reboots for the first time.

You can then proceed with *Chapter 5, Installing Using Anaconda*.

11.2.3. Installing in VNC Connect Mode

VNC connect mode is when the system being installed initiates a connection to the VNC viewer running on a remote system. Before you start, make sure the remote system is configured to accept incoming connection on the port you want to use for VNC. The exact way to make sure the connection will not be blocked depends on your network and on your workstation's configuration. Information about configuring the firewall in Fedora is available in the *Fedora Security Guide*, available at *http://docs.fedoraproject.org/*.

Procedure 11.2. Starting VNC in Connect Mode

1. Start the VNC viewer on the client system in listening mode. For example, on Fedora using **TigerVNC**, execute the following command:

   ```
   $ vncviewer -listen PORT
   ```

   Replace *PORT* with the port number you want to use for the connection.

   The terminal will display a message similar to the following example:

   > Example 11.1. TigerVNC Viewer Listening
   >
   > ```
   > TigerVNC Viewer 64-bit v1.3.0 (20130924)
   > Built on Sep 24 2013 at 16:32:56
   > Copyright (C) 1999-2011 TigerVNC Team and many others (see README.txt)
   > See http://www.tigervnc.org for information on TigerVNC.
   >
   > Thu Feb 20 15:23:54 2014
   > main:        Listening on port 5901
   > ```

   When this message is displayed, the VNC viewer is ready and waiting for an incoming connection from the installation system.

2. Boot the installation system and wait for the boot menu to appear. In the menu, edit boot options (see *Section 4.2, "The Boot Menu"*) and append the following options to the end of the command line:

   ```
   inst.vnc inst.vncconnect=HOST:PORT
   ```

   Replace *HOST* with the IP address of the system running the listening VNC viewer, and *PORT* with the port number that the VNC viewer is listening on.

3. Start the installation. The system will initialize the installation program and start the necessary services. Once the initialization is finished, **Anaconda** will attempt to connect to the IP address and port you provided in the previous step.

   When the connection is successfully established, a new window will open on the system running the VNC viewer, displaying the installation menu. This window will provide full remote access to the installer until the installation finishes and the system reboots for the first time.

You can then proceed with *Chapter 5, Installing Using Anaconda*.

11.3. Kickstart Considerations

Commands for using a VNC installation are also available in Kickstart installations. Using just the **vnc** command will set up an installation using Direct Mode. Options are available to set up an installation using Connect Mode. For more information about the **vnc** command and options used in Kickstart files, see *Appendix A, Kickstart Syntax Reference*.

11.4. Considerations for Headless Systems

When installing headless systems, the only choices are an automated Kickstart installation or an interactive VNC installation using connect mode. For more information about automated Kickstart installation, see *Appendix A, Kickstart Syntax Reference*. The general process for an interactive VNC installation is described below.

1. Set up a PXE server that will be used to start the installation. Information about installing and performing basic configurating of a PXE server can be found in *Chapter 10, Setting Up an Installation Server*.

2. Configure the PXE server to use the boot options for a connect mode VNC installation. For information on these boot options, see *Section 11.2.3, "Installing in VNC Connect Mode"*.

3. Follow the procedure for a VNC Installation using connect mode as described in the *Procedure 11.2, "Starting VNC in Connect Mode"*. However, when directed to boot the system, boot it from the PXE server.

Upgrading Your Current System

This chapter explains how to upgrade your existing Fedora installation to the current release. There are two basic ways to do so:

Automatic upgrade using **FedUp**

The preferred way to upgrade your system is an automatic upgrade using the **FedUp** utility. For information on performing an automatic upgrade, see *Section 12.1, "Automatic System Upgrade Using FedUp"*.

Manual Reinstallation

You can upgrade to the latest version of Fedora manually instead of relying on **FedUp**. This involves booting the installer as if you were performing a clean installation, letting it detect your existing Fedora system, and overwriting the root partition while preserving data on other partitions and volumes. The same process can also be used to reinstall the system, if you need to. For detailed information, see *Section 12.2, "Manual System Upgrade or Reinstallation"*.

Warning

Always back up your data before performing an upgrade or reinstalling your system, no matter which method you choose.

12.1. Automatic System Upgrade Using FedUp

Unfortunately, we have not written this chapter yet. In the meantime before we fix this, you can follow the instructions at *https://fedoraproject.org/wiki/FedUp*.

12.2. Manual System Upgrade or Reinstallation

Unfortunately, we have not written this chapter yet, and there is no dedicated documentation about a manual reinstall on the Wiki, either. In the meantime before we fix this, you can try to start the installation normally (from a boot CD/DVD/USB), select manual partitioning in your installer, and reuse existing partitions instead of destroying them and creating new ones. The instructions at *Section 5.4.10, "Manual Partitioning"* should in most cases be easy to adapt for this.

Part III. Technical Appendixes

The following appendixes do not contain instructions that tell you how to install Fedora. Instead, they provide technical background that you might find helpful to understand the options that Fedora offers you at various points in the installation process.

Appendix A. Kickstart Syntax Reference

This appendix describes commands and options available in Kickstart installations. For general information about Kickstart, see *Chapter 9, Automating the Installation with Kickstart*.

Important

Device names are not guaranteed to be consistent across reboots, which can complicate usage in Kickstart scripts. When a Kickstart option calls for a device node name (such as **sda**), you can instead use any item from **/dev/disk**. For example, instead of:

```
part / --fstype=xfs --onpart=sda1
```

You could use an entry similar to one of the following:

```
part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

This provides a consistent way to refer to disks that is more meaningful than just **sda**. This is especially useful in large storage environments.

While the general principles of Kickstart installations tend to stay the same, the commands and options can change between major releases. You can use the **ksverdiff** command to display the differences between two versions of the Kickstart syntax. This is useful when updating an existing Kickstart file to be used with a new release. To display a list of changes in syntax between Fedora 21 and 22, use the following command:

```
$ ksverdiff -f F21 -t F22
```

The **-f** option specifies the release to start the comparison with, and the **-t** option to specify the release to end with. For additional information, see the `ksverdiff(1)` man page. Also note that you can not use this to display changes in a release that is newer than your system - the version of *pykickstart* on Fedora 21 can not display changes in Fedora 22.

Additionally, you can review the *Fedora 22 Release Notes*, available at http://docs.fedoraproject.org/, for a list of changes.

Note

In the following sections, if an option is followed by an equals mark (**=**), a value must be specified after it. In the example commands, options in square brackets (**[ ]**) are optional arguments for the command.

A.1. Installation Methods and Sources

The following commands control the way Fedora will be installed.

A.1.1. device (optional) - Install Extra Device Drivers

On most PCI systems, the installation program will automatically detect Ethernet and SCSI cards. However, on older systems and some PCI systems, Kickstart requires a hint to find the proper devices. The **`device`** command, which tells the installation program to install extra modules, uses the following format:

```
device moduleName [--opts=]
```

Replace *`moduleName`* with the name of the kernel module which should be installed.

`--opts=`

Options to pass to the installed kernel module. For example:

```
device i2c_piix4 --opts="aic152x=0x340 io=11"
```

A.1.2. driverdisk (optional) - Use a Driver Disk

Driver disks can be used during Kickstart installations to provide additional drivers not included by default. You must copy the driver disks's contents to the root directory of a partition on the system's hard drive. Then, you must use the **`driverdisk`** command to specify that the installation program should look for a driver disk and its location.

```
driverdisk partition | --source= | --biospart=
```

`partition`

Search for the driver disk image on a local partition. Replace *`partition`* with the name of the partition containing the driver disk. Note that the partition must be specified as a full path. For example:

```
driverdisk /dev/sdb1
```

`--source=`

Search for the driver disk in a network location instead of a local partition. For example:

```
driverdisk --source=ftp://path/to/dd.img
```

```
driverdisk --source=http://path/to/dd.img
```

```
driverdisk --source=nfs:hostname:/path/to/dd.img
```

`--biospart=`

BIOS partition containing the driver disk (for example, **`82p2`**).

A.1.3. install (required) - Configure Installation Method

The default installation mode. You must specify the type of installation from **`cdrom`**, **`harddrive`**, **`nfs`**, **`liveimg`**, or **`url`**. The **`install`** command and the installation method command must be on separate lines. For example:

```
install
liveimg --url=file:///images/install/squashfs.img --noverifyssl
```

The installation method commands are:

`cdrom`

Install from the first optical (DVD) drive on the system.

`harddrive`

Install from a tree or full installation ISO image on a local hard drive. The tree or ISO image must be on a file system which is mountable in the installation environment. Supported file systems are `ext2`, `ext3`, `ext4`, `vfat`, or `xfs`.

```
install
harddrive --partition= | --biospart= [--dir=]
```

`--partition=`

Partition to install from (such as **`sdb2`**).

`--biospart=`

BIOS partition to install from (such as **`82p2`**).

`--dir=`

Directory containing the installation tree or ISO image.

`liveimg`

Install from a disk image instead of packages. The image can be the **`squashfs.img`** file from a live ISO image, or any file system that the installation media can mount. Supported file systems are `ext2`, `ext3`, `ext4`, `vfat`, and `xfs`.

This command also supports installation from tar archives of the root file system. In that case, the file name must end with **`.tar`**, **`.tbz`**, **`.tgz`**, **`.txz`**, **`.tar.bz2`**, **`tar.gz`**, or **`tar.xz`**.

```
install
liveimg --url= [--proxy= | --checksum= | --noverifyssl=]
```

`--url=`

The location to install from. Supported protocols are `HTTP`, `HTTPS`, `FTP`, and `file`.

`--proxy=`

Specify an `HTTP`, `HTTPS` or `FTP` proxy to use while performing the installation.

`--checksum=`

An optional argument with the `SHA256` checksum of the image file, used for integrity verification. If you are using a live image provided by Fedora Project, you can find a list of checksums at *https://fedoraproject.org/en/verify*.

`--noverifyssl`
Disable SSL verification when connecting to an HTTPS server.

`nfs`
Install from an NFS server specified. The NFS server must be exporting the full installation ISO image (such as the Fedora Server DVD) or its extracted contents.

```
install
nfs --server= [--dir=] [--opts= ]
```

`--server=`
Host name of the server.

`--dir=`
Directory containing the installation tree or ISO image.

`--opts=`
Mount options to use for mounting the NFS export.

`url`
Install from a tree on a remote server via HTTP, HTTPS, or FTP.

```
install
url --url= | --mirrorlist= [--proxy= | --noverifyssl]
```

`--url=`
The location to install from. Supported protocols are **`http`**, **`https`**, **`ftp`**, and **`file`**.

`--mirrorlist=`
The mirror URL to install from.

`--proxy=`
Specify an HTTP, HTTPS or FTP proxy to use while performing the installation.

`--noverifyssl`
Disable SSL verification when connecting to an HTTPS server.

A.1.4. mediacheck (optional) - Verify Installation Media Integrity

This command will force the installation program to perform a media check before starting the installation, similarly to the **`rd.live.check`** boot option (see *Section 8.5.2, “Verifying Boot Media”*. This command requires that installations be attended, so it is disabled by default.

A.1.5. ostreesetup (optional) - Install from an OSTree

Used for OSTree installations. See *https://wiki.gnome.org/action/show/Projects/OSTree* for more information about OSTree. Available options are:

`--osname=`
Management root for OS installation (required).

`--remote=`
Management root for OS installation (optional).

--url=
: Repository URL (required).

--ref=
: Name of branch inside the repository (required).

--nogpgcheck
: Disable GPG key verification (optional).

A.1.6. repo (optional) - Configure Additional Repositories

Configures additional **DNF** repositories that may be used as sources for package installation. This command can be used multiple times in a single Kickstart file.

See the *Fedora System Administrator's Guide*, available at http://docs.fedoraproject.org/, for information about the **DNF** package manager.

> **Important**
>
> Repositories used for installation must be stable. The installation may fail if a repository is modified before the installation concludes.

--name=
: The repository ID. This option is required. If a repository has a name which conflicts with another previously added repository, it will be ignored. Because the installation program uses a list of pre-configured repositories, this means that you cannot add repositories with the same names as the preconfigured ones.

--baseurl=
: The repository URL. The variables that may be used in **DNF** repo configuration files are *not* supported. You may use one of either this option or **--mirrorlist**, not both.

--mirrorlist=
: The URL pointing at a list of mirrors for the repository. The variables that may normally be used in dnf repository configuration files are not supported here. You may use one of either this option or --baseurl, not both.

--install
: Make the repository configured in the Kickstart file available on the system after the installation as well. Creates a configuration file for the repository in **/etc/yum.repos.d/** on the installed system.

--cost=
: An integer value to assign a cost to this repository. If multiple repositories provide the same packages, this number will be used to prioritize which repository will be used before another. Repositories with a lower cost take priority over repositories with higher cost.

--excludepkgs=
: A comma-separated list of package names that must *not* be pulled from this repository. This is useful if multiple repositories provide the same package and you want to make sure it comes from a particular repository. Both full package names (such as **publican**) and globs (such as **gnome-****) are accepted.

`--includepkgs=`
: A comma-separated list of package names and globs that must be pulled from this repository. This is useful if multiple repositories provide the same package and you want to make sure it comes from this repository.

`--proxy=`
: Specify an HTTP, HTTPS or FTP proxy server to use when accessing this repository. This setting does not affect any other repositories or installation sources.

`--ignoregroups=true`
: This option is used when composing installation trees and has no effect on the installation process itself. It tells the compose tools to not look at the package group information when mirroring trees so as to avoid mirroring large amounts of unnecessary data.

`--noverifyssl`
: Disable SSL verification when connecting to an HTTPS server.

A.2. Storage and Partitioning

Commands in this section are used to determine your system's storage options and partitioning.

A.2.1. autopart (optional) - Automatic Partitioning

Automatically creates partitions: a root (`/`) partition (1 GB or larger), a **`swap`** partition, and an appropriate **`/boot`** partition for the architecture. On large enough drives (50 GB and larger), this also creates a **`/home`** partition.

> **Important**
>
> The **`autopart`** option cannot be used together with the **`part/partition`**, **`raid`**, **`logvol`**, or **`volgroup`** options in the same Kickstart file.

`--type=`
: Selects one of the predefined automatic partitioning schemes you want to use. Accepts the following values:

- **`lvm`**: The LVM partitioning scheme.
- **`btrfs`**: The Btrfs partitioning scheme.
- **`plain`**: Regular partitions with no LVM or Btrfs.
- **`thinp`**: The LVM Thin Provisioning partitioning scheme.

The created partitioning scheme will follow the recommended scheme described at *Section 5.4.10.6, “Recommended Partitioning Scheme”*.

`--fstype=`
: Specify a supported file system (such as **`ext4`** or **`xfs`**) to replace the default when doing automatic partitioning.

`--nolvm`
: Do not use LVM or Btrfs for automatic partitioning. This option is equal to **`--type=plain`**.

`--encrypted`
: Encrypts all partitions. This is equivalent to checking the **Encrypt partitions** check box on the initial partitioning screen during a manual graphical installation.

`--passphrase=`
: Provides a default system-wide passphrase for all encrypted devices.

`--escrowcert=`*`URL_of_X.509_certificate`*
: Stores data encryption keys of all encrypted volumes as files in **`/root`**, encrypted using the X.509 certificate from the URL specified with *`URL_of_X.509_certificate`*. The keys are stored as a separate file for each encrypted volume. This option is only meaningful if **`--encrypted`** is specified.

`--backuppassphrase`
: Adds a randomly-generated passphrase to each encrypted volume. Store these passphrases in separate files in **`/root`**, encrypted using the X.509 certificate specified with **`--escrowcert`**. This option is only meaningful if **`--escrowcert`** is specified.

`--cipher=`
: Specifies which type of encryption will be used if the **Anaconda** default `aes-xts-plain64` is not satisfactory. You must use this option together with the **`--encrypted`** option; by itself it has no effect. Available types of encryption are listed in the *Fedora Security Guide*, available at *http://docs.fedoraproject.org/*. Using either `aes-xts-plain64` or `aes-cbc-essiv:sha256` is strongly recommended.

A.2.2. bootloader (required) - Configure Boot Loader

Specifies how the boot loader should be installed.

> **Important**
>
> You should always use a password to protect your boot loader. An unprotected boot loader can allow a potential attacker to modify the system's boot options and gain unauthorized access to the system.

> **Important**
>
> Some systems require a special partition for installing the boot loader. The type and size of this partition depends on whether the disk you are installing the boot loader to uses the *Master Boot Record* (MBR) or a *GUID Partition Table* (GPT) schema. For more information, see *Section 5.4.8.1, “Boot Loader Installation”*.

`--append=`
: Specifies additional kernel parameters. To specify multiple parameters, separate them with spaces. For example:

```
bootloader --location=mbr --append="hdd=ide-scsi ide=nodma"
```

The **rhgb** and **quiet** parameters are always used, even if you do not specify them here or do not use the **--append=** command at all.

--boot-drive=
Specifies which drive the boot loader should be written to, and therefore which drive the computer will boot from. If you use a multipath device as the boot drive, specify only one member of the device.

> **Important**
>
> The **--boot-drive=** option is currently being ignored in Fedora installations on IBM System z systems using the **zipl** boot loader. When **zipl** is installed, it determines the boot drive on its own.

--leavebootloader
Prevents the installation program from making changes to the existing list of bootable images on UEFI or ISeries/PSeries systems.

--driveorder=
Specifies which drive is first in the BIOS boot order. For example:

```
bootloader --driveorder=sda,hda
```

--location=
Specifies where the boot record is written. Valid values are the following:

- **mbr** - The default option. Depends on whether the drive uses the Master Boot Record (MBR) or GUID Partition Table (GPT) scheme:
  - On a GPT-formatted disk, this option will install stage 1.5 of the boot loader into the BIOS boot partition.
  - On an MBR-formatted disk, stage 1.5 will be installed into the empty space between the MBR and the first partition.
- **partition** - Install the boot loader on the first sector of the partition containing the kernel.
- **none** - Do not install the boot loader.

In most cases, this option does not need to be specified.

--password=
If using **GRUB2** as the boot loader, sets the boot loader password to the one specified with this option. This should be used to restrict access to the **GRUB2** shell, where arbitrary kernel options can be passed.

If a password is specified, **GRUB2** will also ask for a user name. The user name is always **root**.

`--iscrypted`

Normally, when you specify a boot loader password using the **`--password=`** option, it will be stored in the Kickstart file in plain text. If you want to encrypt the password, use this option and an encrypted password.

To generate an encrypted password, use the **`grub2-mkpasswd-pbkdf2`** command, enter the password you want to use, and copy the command's output (the hash starting with **`grub.pbkdf2`**) into the Kickstart file. An example **`bootloader`** Kickstart entry with an encrypted password will look similar to the following:

```
bootloader --iscrypted --
password=grub.pbkdf2.sha512.10000.5520C6C9832F3AC3D149AC0B24BE69E2D4FB0DBEEDBD29CA1D30A044DE2645C4C7A
```

`--timeout=`

Specifies the amount of time the boot loader will wait before booting the default option (in seconds).

`--default=`

Sets the default boot image in the boot loader configuration.

`--extlinux`

Use the **extlinux** boot loader instead of **GRUB2**. This option only works on systems supported by **extlinux**.

`--disabled`

Do not attempt to install a boot loader. This option overrides all other boot loader configuration; all other boot loader options will be ignored and no boot loader packages will be installed.

A.2.3. btrfs (optional) - Create Btrfs Volume or Subvolume

Create a Btrfs volume or subvolume. For a volume, the syntax is:

```
btrfs mntpoint --data=level --metadata=level [--label=] partitions
```

One or more partitions can be specified in *`partitions`*. When specifying more than one partitions, the entries must be separated by a single space. See *Example A.1, “Creating Btrfs Volumes and Subvolumes”* for a demonstration.

For a subvolume, the syntax is:

```
btrfs mntpoint --subvol --name=name parent
```

`parent` should be the identifier of the subvolume's parent volume, *name* with a name for the subvolume, and *`mntpoint`* is the location where the file system is mounted.

`--data=`

RAID level to use for file system *data* (such as **`0`**, **`1`**, or **`10`**). This parameter is optional, has no meaning for subvolumes, and requires more than one physical disk.

`--metadata=`

RAID level to use for file system/volume *metadata* (such as **`0`**, **`1`**, or **`10`**). This parameter is optional, has no meaning for subvolumes, and requires more than one physical disk.

`--label=`
: Specify a label for the Btrfs file system. If the given label is already in use by another file system, a new label will be created. This option has no meaning for subvolumes.

`--subvol`
: Create a Btrfs subvolume instead of a volume.

`--name=`
: Set a name for a Btrfs subvolume.

`--noformat` or **`--useexisting`**
: Use an existing Btrfs volume (or subvolume) and do not reformat the file system.

The following example shows how to create a Btrfs volume from member partitions on three disks with subvolumes for **`/`** and **`/home`**. The main volume is not mounted or used directly in this example.

> Example A.1. Creating Btrfs Volumes and Subvolumes
>
> ```
> part btrfs.01 --size=6000 --ondisk=sda
> part btrfs.02 --size=6000 --ondisk=sdb
> part btrfs.03 --size=6000 --ondisk=sdc
>
> btrfs none --data=0 --metadata=1 --label=f22 btrfs.01 btrfs.02 btrfs.03
> btrfs / --subvol --name=root LABEL=f22
> btrfs /home --subvol --name=home f22
> ```

A.2.4. clearpart (optional) - Remove All Existing Partitions

Removes partitions from the system, prior to creation of new partitions. By default, no partitions are removed.

> **Note**
>
> If the **`clearpart`** command is used, then the **`part --onpart`** command cannot be used on a logical partition.

For a detailed example of partitioning including the **`clearpart`** command, see *Section A.15.1, “Advanced Partitioning Example”*.

`--all`
: Erases all partitions from the system.

> **Warning**
>
> This option will erase all disks which can be reached by the installer, including any attached network storage. Use this option with caution.

--drives=

Specifies which drives to clear partitions from. For example, the following clears all the partitions on the first two drives on the primary IDE controller:

```
clearpart --drives=hda,hdb --all
```

To clear a multipath device, use the format **disk/by-id/scsi-*WWID***, where *WWID* is the *world-wide identifier* for the device. For example, to clear a disk with WWID **58095BEC5510947BE8C0360F604351918**, use:

```
clearpart --drives=disk/by-id/scsi-58095BEC5510947BE8C0360F604351918
```

This format is preferable for all multipath devices, but if errors arise, multipath devices that do not use *logical volume management* (LVM) can also be cleared using the format **disk/by-id/dm-uuid-mpath-*WWID***, where *WWID* is the *world-wide identifier* for the device. For example, to clear a disk with WWID **2416CD96995134CA5D787F00A5AA11017**, use:

```
clearpart --drives=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```

Warning

Never specify multipath devices by device names like **mpatha**. Device names such as this are not specific to a particular disk. The disk named **/dev/mpatha** during installation might not be the one that you expect it to be. Therefore, the **clearpart** command could target the wrong disk.

--list=

Specifies which partitions to clear. This option overrides the **--all** and **--linux** options if used. Can be used across different drives. For example:

```
clearpart --list=sda2,sda3,sdb1
```

--disklabel=

Create a set disk label when relabeling a disk.

--linux

Erases all Linux partitions.

--none

Do not remove any partitions. This is the default behavior - using this option is the same as not using the **clearpart** command at all.

Note

Using the **`clearpart --all`** command in a Kickstart file to remove all existing partitions during the installation will cause **Anaconda** to pause and prompt you for a confirmation. If you need to perform the installation automatically with no interaction, add the **`zerombr`** command to your Kickstart file.

A.2.5. fcoe (optional) - Configure Fibre Channel Over Ethernet Devices

Specify which FCoE devices should be activated automatically in addition to those discovered by *Enhanced Disk Drive Services* (EDD).

```
fcoe --nic=name [--dcp= | --autovlan]
```

`--nic=` (required)
Name of the device to be activated.

`--dcb=`
Establish *Data Center Bridging* (DCB) settings.

`--autovlan`
Discover VLANs automatically.

A.2.6. ignoredisk (optional) - Ignore Specified Disks

Causes the installation program to ignore the specified disks. This is useful if you use autopartition and want to be sure that some disks are ignored. For example, without **`ignoredisk`**, attempting to deploy on a SAN cluster the Kickstart would fail, as the installation program detects passive paths to the SAN that return no partition table.

```
ignoredisk --drives= | --only-use= [--interactive]
```

`--drives=`
Specify one or more drives to ignore. Multiple drives can be specified as a comma-separated list. For example:

```
ignoredisk --drives=sda,sdc
```

To ignore a multipath device that does not use *logical volume management* (LVM), use the format **`disk/by-id/dm-uuid-mpath-`*`WWID`***, where *`WWID`* is the *world-wide identifier* for the device. For example, to ignore a disk with WWID **`2416CD96995134CA5D787F00A5AA11017`**, use:

```
ignoredisk --drives=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```

Multipath devices that use LVM are not assembled until after **Anaconda** has parsed the Kickstart file. Therefore, you cannot specify these devices in the format **`dm-uuid-mpath`**. Instead, to ignore a multipath device that uses LVM, use the format **`disk/by-id/scsi-`*`WWID`***, where

`WWID` is the *world-wide identifier* for the device. For example, to ignore a disk with WWID **`58095BEC5510947BE8C0360F604351918`**, use:

```
ignoredisk --drives=disk/by-id/scsi-58095BEC5510947BE8C0360F604351918
```

Warning

Never specify multipath devices by device names like **`mpatha`**. Device names such as this are not specific to a particular disk. The disk named **`/dev/mpatha`** during installation might not be the one that you expect it to be. Therefore, the **`clearpart`** command could target the wrong disk.

`--only-use=`

Specifies a list of disks for the installation program to use. All other disks are ignored. For example, to use disk **`sda`** during installation and ignore all other disks:

```
ignoredisk --only-use=sda
```

To include a multipath device that does not use LVM:

```
ignoredisk --only-use=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```

To include a multipath device that uses LVM:

```
ignoredisk --only-use=disk/by-id/scsi-58095BEC5510947BE8C0360F604351918
```

`--interactive`

Allows you to manually navigate the advanced storage screen.

A.2.7. iscsi (optional) - Configure iSCSI Devices

Specifies additional iSCSI storage to be attached during installation. If you use the **`iscsi`** command, you must also assign a name to the iSCSI node, using the **`iscsiname`** command (see *Section A.2.8, “iscsiname (optional) - Assign Name to iSCSI Device”*. The **`iscsiname`** command must appear before the **`iscsi`** command in the Kickstart file.

You should configure iSCSI storage in the system BIOS or firmware (iBFT for Intel systems) rather than use the **`iscsi`** command if possible. If you do so, **Anaconda** automatically detects and uses disks configured in BIOS or firmware and no special configuration is necessary in the Kickstart file.

If you must use the **`iscsi`** command, make sure that networking is activated at the beginning of the installation, and that the **`iscsi`** command appears in the Kickstart file *before* you refer to iSCSI disks with commands such as **`clearpart`** or **`ignoredisk`**.

```
iscsi --ipaddr= --port= [--target= | --iface= | --user= | --password= | --reverse-user= | --
reverse-password=]
```

`--ipaddr=`

The IP address of the target to connect to.

`--port=`
The port number (typically **`3260`**).

`--target=`
Target *IQN* (iSCSI Qualified Name).

`--iface=`
Bind the connection to a specific network interface instead of using the default one determined by the network layer. Once used, it must be specified in all instances of the **`iscsi`** command in the entire Kickstart file.

`--user=`
User name required to authenticate with the target.

`--password=`
Password that corresponds with the user name specified for the target.

`--reverse-user=`
User name required to authenticate with the initiator from a target using reverse CHAP authentication.

`--reverse-password=`
Password that corresponds with the user name specified for the initiator.

A.2.8. iscsiname (optional) - Assign Name to iSCSI Device

Assigns a name to an iSCSI node specified by the **`iscsi`** command (*Section A.2.7, “iscsi (optional) - Configure iSCSI Devices”*). This command is mandatory if you use the **`iscsi`** command, and it must be specified *before* you use **`iscsi`**.

```
iscsiname iqn
```

A.2.9. logvol (optional) - Create LVM Logical Volume

Create a logical volume for Logical Volume Management (LVM) with the syntax:

```
logvol mntpoint --vgname= --name= [options]
```

Note

Do not use the dash (**`-`**) character in logical volume and volume group names when installing Fedora using Kickstart. If this character is used, the installation will finish normally, but the **`/dev/mapper/`** directory will list these volumes and volume groups with every dash doubled. For example, a volume group named **`volgrp-01`** containing a logical volume named **`logvol-01`** will be listed as **`/dev/mapper/volgrp--01-logvol--01`**.

This limitation only applies to newly created logical volume and volume group names. If you are reusing existing ones using the **`--noformat`** or **`--useexisting`** option, their names will not be changed.

For a detailed example of **`logvol`** in action, see *Section A.15.1, “Advanced Partitioning Example”*.

mntpoint

Replace with the volume's mount point. This name can take the following forms:

/path

A path to the mount point - for example, **/** or **/home**

swap

The partition is used as swap space.

To determine the size of the swap partition automatically, use the **--recommended** option:

```
swap --recommended
```

To determine the size of the swap partition automatically but also allow extra space for your system to hibernate, use the **--hibernation** option:

```
swap --hibernation
```

The size assigned will be equivalent to the swap space assigned by **--recommended** plus the amount of RAM on your system.

For the swap sizes assigned by these commands, see the section describing swap in *Section 5.4.10.6, "Recommended Partitioning Scheme"*.

none

Used only when creating a thin pool volume.

--noformat

Use an existing logical volume and do not format it.

--useexisting

Use an existing logical volume and format it.

--fstype=

Sets the file system type for the logical volume. Valid values are **xfs**, **ext2**, **ext3**, **ext4**, **swap**, and **vfat**. See *Section 5.4.10.5, "Device, File System and RAID Types"* for information about available file systems.

--fsoptions=

Specifies a free form string of options to be used when mounting the filesystem. This string will be copied into the **/etc/fstab** file of the installed system and should be enclosed in quotes. For example:

```
--fsoptions="ro, x-systemd.device-timeout=0"
```

--label=

Sets a label for the logical volume.

--grow

Grow the volume to fill available space (if any), or up to the limit set by the **--maxsize=** option.

--size=

The size of the logical volume in megabytes. This option can not be used together with the **--percent=** option.

`--percent=`

Specify the amount by which to grow the logical volume, as a percentage of the free space in the volume group after any statically-sized logical volumes are taken into account. This option can not be used together with the **`--size=`** and **`--grow`** options.

> **Important**
>
> When creating a new logical volume, you must either specify its size statically using the **`--size=`** option, or as a percentage of remaining free space using the **`--percent=`** option. You can not use both of these options on the same logical volume.

`--maxsize=`

The maximum size in megabytes when the logical volume is set to grow. Specify an integer value here such as **`500`** (do not include the unit).

`--recommended`

Use this option when creating a `swap` logical volume to determine the size of this volume automatically, based on your system's hardware. For details about the recommended scheme, see *Section 5.4.10.6, “Recommended Partitioning Scheme”*.

`--resize`

Resize an existing logical volume. If you use this option, you must also specify **`--useexisting`** and **`--size`**.

`--encrypted`

Specifies that this logical volume should be encrypted, using the passphrase provided in the **`--passphrase=`** option. If you do not specify a passphrase, the installation program will use the default, system-wide passphrase set with the **`autopart --passphrase`** command, or stop the installation and prompt you to provide a passphrase if no default is set.

`--passphrase=`

Specifies the passphrase to use when encrypting this logical volume. You must use this option together with the **`--encrypted`** option. This option has no effect by itself.

`--cipher=`

Specifies which type of encryption will be used if the **Anaconda** default `aes-xts-plain64` is not satisfactory. You must use this option together with the **`--encrypted`** option; by itself it has no effect. Available types of encryption are listed in the *Fedora Security Guide*, available at *http://docs.fedoraproject.org/*. Using either `aes-xts-plain64` or `aes-cbc-essiv:sha256` is strongly recommended.

`--escrowcert=`*`URL_of_X.509_certificate`*

Store data encryption keys of all encrypted volumes as files in **`/root`**, encrypted using the X.509 certificate from the URL specified with *`URL_of_X.509_certificate`*. The keys are stored as a separate file for each encrypted volume. This option is only meaningful if **`--encrypted`** is specified.

`--backuppassphrase`

Add a randomly-generated passphrase to each encrypted volume. Store these passphrases in separate files in **`/root`**, encrypted using the X.509 certificate specified with **`--escrowcert`**. This option is only meaningful if **`--escrowcert`** is specified.

`--thinpool`
Creates a thin pool logical volume. (Use a mount point of **none**)

`--metadatasize=`
Metadata area size (in MiB) for a new thin pool device.

`--chunksize=`
Chunk size (in KiB) for a new thin pool device.

`--thin`
Create a thin logical volume. (Requires use of **`--poolname`**)

`--poolname=`
Specify the name of the thin pool in which to create a thin logical volume. Requires the **`--thin`** option.

`--profile=`
Specify the configuration profile name to use with thin logical volumes. If used, the name will also be included in the metadata for the given logical volume. By default, the available profiles are **`default`** and **`thin-performance`** and are defined in the **`/etc/lvm/profile`** directory. See the `lvm(8)` man page for additional information.

Create one or more partitions first using *Section A.2.10, “part (required) - Create Physical Partition”*, create the logical volume group (*Section A.2.12, “volgroup (optional) - Create LVM Volume Group”*), and then create logical volumes. For example:

```
part pv.01 --size 3000
volgroup myvg pv.01
logvol / --vgname=myvg --size=2000 --name=rootvol
```

A.2.10. part (required) - Create Physical Partition

Creates a partition on the system.

For a detailed example of **`part`** in action, see *Section A.15.1, “Advanced Partitioning Example”*.

```
part|partition mntpoint --name=name --device=device --rule=rule [options]
```

Warning

All partitions created are formatted as part of the installation process unless **`--noformat`** and **`--onpart=`** are used.

Note

If partitioning fails for any reason, diagnostic messages appear on virtual console 3.

`mntpoint`

Where the partition is mounted. The value must be of one of the following:

`/path`

A path to the mount point - for example, **`/`** or **`/home`**

`swap`

The partition is used as swap space.

To determine the size of the swap partition automatically, use the **`--recommended`** option:

```
swap --recommended
```

The size assigned will be effective but not precisely calibrated for your system.

To determine the size of the swap partition automatically but also allow extra space for your system to hibernate, use the **`--hibernation`** option:

```
swap--hibernation
```

The size assigned will be equivalent to the swap space assigned by **`--recommended`** plus the amount of RAM on your system.

For the swap sizes assigned by these commands, see the section describing swap in *Section 5.4.10.6, "Recommended Partitioning Scheme"*.

`raid.`*`id`*

The partition is used for software RAID (see **`raid`**).

`pv.`*`id`*

The partition is used for LVM (see *Section A.2.10, "part (required) - Create Physical Partition"*).

`biosboot`

The partition will be used for a BIOS Boot partition. A 1 MB BIOS boot partition is necessary on BIOS-based systems using a *GUID Partition Table* (GPT); the boot loader will be installed into it. It is not necessary on UEFI systems. Also see *Section A.2.10, "part (required) - Create Physical Partition"*.

`/boot/efi`

An EFI System Partition. An EFI partition at least 50 MB in size is necessary on UEFI-based systems; the recommended size is 200 MB. It is not necessary on BIOS systems. Also see *Section A.2.10, "part (required) - Create Physical Partition"*.

`--size=`

The minimum partition size in megabytes. Specify an integer value here such as **`500`** (do not include the unit).

> **Important**
>
> If the **`--size`** value is too small, the installation will fail. Set the **`--size`** value as the minimum amount of space you require. For size recommendations, see *Section 5.4.10.6, “Recommended Partitioning Scheme”*.

`--maxsize=`

The maximum partition size in megabytes when the partition is set to grow. Specify an integer value here such as **`500`** (do not include the unit).

`--resize`

Resize an existing partition. When using this option, specify the new size (in megabytes) using the **`--size=`** option and the target partition using the **`--onpart=`** option.

`--grow`

Tells the partition to grow to fill available space (if any), or up to the maximum size setting.

> **Note**
>
> If you use **`--grow=`** without setting **`--maxsize=`** on a swap partition, **Anaconda** will limit the maximum size of the swap partition. For systems that have less than 2 GB of physical memory, the imposed limit is twice the amount of physical memory. For systems with more than 2 GB, the imposed limit is the size of physical memory plus 2 GB.

`--noformat`

Specifies that the partition should not be formatted, for use with the **`--onpart`** command.

`--onpart=` or **`--usepart=`**

Specifies the device on which to place the partition. For example:

```
partition /home --onpart=hda1
```

The above puts **`/home`** on **`/dev/hda1`**.

These options can also add a partition to a logical volume. For example:

```
partition pv.1 --onpart=hda2
```

The device must already exist on the system; the **`--onpart`** option will not create it.

`--ondisk=` or **`--ondrive=`**

Forces the partition to be created on a particular disk. For example, **`--ondisk=sdb`** puts the partition on the second SCSI disk on the system.

To specify a multipath device that does not use *logical volume management* (LVM), use the format **`disk/by-id/dm-uuid-mpath-`*`WWID`***, where *`WWID`* is the *world-wide identifier* for the device. For example, to specify a disk with WWID **`2416CD96995134CA5D787F00A5AA11017`**, use:

```
part / --fstype=xfs --grow --asprimary --size=8192 --ondisk=disk/by-id/dm-uuid-
mpath-2416CD96995134CA5D787F00A5AA11017
```

Multipath devices that use LVM are not assembled until after **Anaconda** has parsed the Kickstart file. Therefore, you cannot specify these devices in the format **`dm-uuid-mpath`**. Instead, to specify a multipath device that uses LVM, use the format **`disk/by-id/scsi-`*`WWID`***, where *`WWID`* is the *world-wide identifier* for the device. For example, to specify a disk with WWID **`58095BEC5510947BE8C0360F604351918`**, use:

```
part / --fstype=xfs --grow --asprimary --size=8192 --ondisk=disk/by-id/
scsi-58095BEC5510947BE8C0360F604351918
```

Warning

Never specify multipath devices by device names like **`mpatha`**. Device names such as this are not specific to a particular disk. The disk named **`/dev/mpatha`** during installation might not be the one that you expect it to be. Therefore, the **`clearpart`** command could target the wrong disk.

`--asprimary`

Forces the partition to be allocated as a *primary* partition. If the partition cannot be allocated as primary (usually due to too many primary partitions being already allocated), the partitioning process will fail. This option only makes sense when the disk uses a Master Boot Record (MBR); for GUID Partition Table (GPT)-labeled disks this option has no meaning. For information about primary (and extended) partitions, see *Section 5.4.10.6, “Recommended Partitioning Scheme”*.

`--fsprofile=`

Specifies a *usage type* to be passed to the program that makes a filesystem on this partition. A usage type defines a variety of tuning parameters to be used when making a filesystem. For this option to work, the filesystem must support the concept of usage types and there must be a configuration file that lists valid types. For **`ext2`**, **`ext3`**, **`ext4`**, this configuration file is **`/etc/mke2fs.conf`**.

`--fstype=`

Sets the file system type for the partition. Valid values are **`xfs`**, **`ext2`**, **`ext3`**, **`ext4`**, **`swap`**, **`vfat`**, **`efi`** and **`biosboot`**. For information about supported file systems, see *Section 5.4.10.5, “Device, File System and RAID Types”*.

`--fsoptions=`

Specifies a free form string of options to be used when mounting the filesystem. This string will be copied into the **`/etc/fstab`** file of the installed system and should be enclosed in quotes. For example:

```
--fsoptions="ro, x-systemd.device-timeout=0"
```

`--label=`

Assign a label to an individual partition.

`--recommended`

Determine the size of the partition automatically. For details about the recommended scheme, see *Section 5.4.10.6, “Recommended Partitioning Scheme”*.

`--onbiosdisk`

Forces the partition to be created on a particular disk as discovered by the BIOS.

`--encrypted`

Specifies that this partition should be encrypted, using the passphrase provided in the **`--passphrase`** option. If you do not specify a passphrase, **Anaconda** uses the default, system-wide passphrase set with the **`autopart --passphrase`** command, or stops the installation and prompts you to provide a passphrase if no default is set.

`--passphrase=`

Specifies the passphrase to use when encrypting this partition. You must use this option together with the **`--encrypted`** option; by itself it has no effect.

`--cipher=`

Specifies which type of encryption will be used if the **Anaconda** default `aes-xts-plain64` is not satisfactory. You must use this option together with the **`--encrypted`** option; by itself it has no effect. Available types of encryption are listed in the *Fedora Security Guide*, available at *http://docs.fedoraproject.org/*. Using either `aes-xts-plain64` or `aes-cbc-essiv:sha256` is strongly recommended.

`--escrowcert=`*`URL_of_X.509_certificate`*

Stores data encryption keys of all encrypted volumes as files in **`/root`**, encrypted using the X.509 certificate from the URL specified with *`URL_of_X.509_certificate`*. The keys are stored as a separate file for each encrypted volume. This option is only meaningful if **`--encrypted`** is specified.

`--backuppassphrase`

Add a randomly-generated passphrase to each encrypted partition. Store these passphrases in separate files in **`/root`**, encrypted using the X.509 certificate specified with **`--escrowcert`**. This option is only meaningful if **`--escrowcert`** is specified.

A.2.11. raid (optional) - Create Software RAID

Assembles a software RAID device. This command is of the form:

```
raid mntpoint --level=level --device=mddevice partitions*
```

For a detailed example of **`raid`** in action, see *Section A.15.1, “Advanced Partitioning Example”*.

`mntpoint`

Location where the RAID file system is mounted. If it is `/`, the RAID level must be 1 unless a boot partition (**`/boot`**) is present. If a boot partition is present, the **`/boot`** partition must be level 1 and the root (`/`) partition can be any of the available types. The *`partitions*`* (which denotes that multiple partitions can be listed) lists the RAID identifiers to add to the RAID array.

`--level=`

RAID level to use (**`0`**, **`1`**, **`4`**, **`5`**, **`6`**, or **`10`**). See *Section 5.4.10.5, “Device, File System and RAID Types”* for information about various RAID types and their requirements.

--device=

Name of the RAID device to use. As of Fedora 22, RAID devices are no longer referred to by names like **md0**. If you have an old (v0.90 metadata) array that you cannot assign a name to, you can specify the array by a filesystem label or UUID (for example, **--device=rhel7-root --label=rhel7-root**).

--spares=

Specifies the number of spare drives allocated for the RAID array. Spare drives are used to rebuild the array in case of drive failure.

--fsprofile=

Specifies a *usage type* to be passed to the program that makes a filesystem on this partition. A usage type defines a variety of tuning parameters to be used when making a filesystem. For this option to work, the filesystem must support the concept of usage types and there must be a configuration file that lists valid types. For **ext2**, **ext3**, **ext4**, this configuration file is **/etc/mke2fs.conf**.

--fstype=

Sets the file system type for the partition. Valid values are **xfs**, **ext2**, **ext3**, **ext4**, **swap**, **vfat**, **efi** and **biosboot**. For information about supported file systems, see *Section 5.4.10.5, “Device, File System and RAID Types”*.

--fsoptions=

Specifies a free form string of options to be used when mounting the filesystem. This string will be copied into the **/etc/fstab** file of the installed system and should be enclosed in quotes. For example:

```
--fsoptions="ro, x-systemd.device-timeout=0"
```

--label=

Specify the label to give to the filesystem to be made. If the given label is already in use by another filesystem, a new label will be created.

--noformat

Use an existing RAID device and do not format it.

--useexisting

Use an existing RAID device and reformat it.

--encrypted

Specifies that this array should be encrypted, using the passphrase provided in the **--passphrase** option. If you do not specify a passphrase, **Anaconda** uses the default, system-wide passphrase set with the **autopart --passphrase** command, or stops the installation and prompts you to provide a passphrase if no default is set.

--passphrase=

Specifies the passphrase to use when encrypting this partition. You must use this option together with the **--encrypted** option; by itself it has no effect.

--cipher=

Specifies which type of encryption will be used if the **Anaconda** default `aes-xts-plain64` is not satisfactory. You must use this option together with the **--encrypted** option; by itself it has no effect. Available types of encryption are listed in the *Fedora Security Guide*, available at http://docs.fedoraproject.org/. Using either `aes-xts-plain64` or `aes-cbc-essiv:sha256` is strongly recommended.

--escrowcert=*URL_of_X.509_certificate*

Stores data encryption keys of all encrypted volumes as files in **/root**, encrypted using the X.509 certificate from the URL specified with *URL_of_X.509_certificate*. The keys are stored as a separate file for each encrypted volume. This option is only meaningful if **--encrypted** is specified.

--backuppassphrase

Add a randomly-generated passphrase to each encrypted partition. Store these passphrases in separate files in **/root**, encrypted using the X.509 certificate specified with **--escrowcert**. This option is only meaningful if **--escrowcert** is specified.

The following example shows how to create a RAID level 1 partition for **/**, and a RAID level 5 for **/home**, assuming there are three SCSI disks on the system. It also creates three swap partitions, one on each drive.

Example A.2. Creating a RAID array in Kickstart

```
part raid.01 --size=6000 --ondisk=sda
part raid.02 --size=6000 --ondisk=sdb
part raid.03 --size=6000 --ondisk=sdc

part swap --size=512 --ondisk=sda
part swap --size=512 --ondisk=sdb
part swap --size=512 --ondisk=sdc

part raid.11 --size=1 --grow --ondisk=sda
part raid.12 --size=1 --grow --ondisk=sdb
part raid.13 --size=1 --grow --ondisk=sdc

raid / --level=1 --device=f22-root --label=f22-root raid.01 raid.02 raid.03
raid /home --level=5 --device=f22-home --label=f22-home raid.11 raid.12 raid.13
```

A.2.12. volgroup (optional) - Create LVM Volume Group

Creates a Logical Volume Management (LVM) volume group.

```
volgroup name partition [options]
```

Important

Do not use the dash (`-`) character in logical volume and volume group names when installing Fedora using Kickstart. If this character is used, the installation will finish normally, but the `/dev/mapper/` directory will list these volumes and volume groups with every dash doubled. For example, a volume group named **`volgrp-01`** containing a logical volume named **`logvol-01`** will be listed as **`/dev/mapper/volgrp--01-logvol--01`**.

This limitation only applies to newly created logical volume and volume group names. If you are reusing existing ones using the **`--noformat`** or **`--noformat`** option, their names will not be changed.

For a detailed partitioning example including **`volgroup`**, see *Section A.15.1, “Advanced Partitioning Example”*.

`--noformat`
: Use an existing volume group and do not format it.

`--useexisting`
: Use an existing volume group and reformat it.

`--pesize=`
: Set the size of the physical extents. The default size for Kickstart installations is 4 MiB.

`--reserved-space=`
: Specify an amount of space to leave unused in a volume group in megabytes. Applicable only to newly created volume groups.

`--reserved-percent=`
: Specify a percentage of total volume group space to leave unused. Applicable only to newly created volume groups.

Create one or more partitions first using *Section A.2.10, “part (required) - Create Physical Partition”*, create the logical volume group (*Section A.2.12, “volgroup (optional) - Create LVM Volume Group”*), and then create logical volumes. For example:

```
part pv.01 --size 3000
volgroup myvg pv.01
logvol / --vgname=myvg --size=2000 --name=rootvol
```

A.2.13. zerombr (optional) - Reinitialize Partition Tables

If **`zerombr`** is specified, any invalid partition tables found on disks are initialized. This destroys all of the contents of disks with invalid partition tables. This command is required when performing an unattended installation on a system with previously initialized disks.

Warning

On IBM System z, if **`zerombr`** is specified, any *Direct Access Storage Device* (DASD) visible to the installation program which is not already low-level formatted is automatically low-level formatted with **dasdfmt**. The command also prevents user choice during interactive installations.

If **`zerombr`** is not specified and there is at least one unformatted DASD visible to the installation program, a non-interactive Kickstart installation will exit unsuccessfully.

If **`zerombr`** is not specified and there is at least one unformatted DASD visible to the installation program, an interactive installation exits if the user does not agree to format all visible and unformatted DASDs. To circumvent this, only activate those DASDs that you will use during installation. You can always add more DASDs after installation is complete.

A.2.14. zfcp (optional) - Configure Fibre Channel Device

Define a Fibre channel device. This option only applies on IBM System z. All of the options described below must be specified.

```
zfcp --devnum=devnum --wwpn=wwpn --fcplun=lun
```

`--devnum`

The device number (zFCP adaptor device bus ID).

`--wwpn`

The device's World Wide Port Name (WWPN). Takes the form of a 16-digit number, preceded by **`0x`**.

`--fcplun`

The device's Logical Unit Number (LUN). Takes the form of a 16-digit number, preceded by **`0x`**.

For example:

```
zfcp --devnum=0.0.4000 --wwpn=0x5005076300C213e9 --fcplun=0x5022000000000000
```

A.3. Network Configuration

Commands in this chapter are used for network configuration.

A.3.1. firewall (optional) - Configure Firewall

Specify the firewall configuration for the installed system.

```
firewall --enabled | --disabled device [--trust= | --ssh | --smtp | --http | --ftp | --port=
 | --service=]
```

`--enabled` or **`--enable`**

Reject incoming connections that are not in response to outbound requests, such as DNS replies or DHCP requests. If access to services running on this machine is needed, you can choose to allow specific services through the firewall.

`--disabled` or **`--disable`**

Disable the firewall.

`--trust=`

Listing a device here, such as **`em1`**, allows all traffic coming to and from that device to go through the firewall. To list more than one device, use this option again - for example:

```
firewall --enable --trust=em1 --trust=em2
```

Do not use a comma-separated format such as **`--trust em1, em2`**.

`incoming`

Replace with one or more of the following to allow the specified services through the firewall:

- **`--ssh`**
- **`--smtp`**
- **`--http`**
- **`--ftp`**

`--port=`

You can specify that ports be allowed through the firewall using the port:protocol format. For example, to allow IMAP access through your firewall, specify **`imap:tcp`**. Numeric ports can also be specified explicitly; for example, to allow UDP packets on port 1234 through, specify **`1234:udp`**. To specify multiple ports, separate them by commas.

`--service=`

This option provides a higher-level way to allow services through the firewall. Some services (like `cups`, `avahi`, etc.) require multiple ports to be open or other special configuration in order for the service to work. You can specify each individual port with the **`--port`** option, or specify **`--service=`** and open them all at once.

Valid options are anything recognized by the **firewall-offline-cmd** program in the *firewalld* package. If `firewalld` is running, **`firewall-cmd --get-services`** will provide a list of known service names.

A.3.2. network (optional) - Configure Network Interfaces

Configures network information for the target system and activates network devices in the installation environment. The device specified in the first **`network`** command is activated automatically. Activation of the device can be also explicitly required by the **`--activate`** option.

`--activate`

If you use the **`--activate`** option on a device that has already been activated (for example, an interface you configured with boot options so that the system could retrieve the Kickstart file) the device is reactivated to use the details specified in the Kickstart file.

Use the **`--nodefroute`** option to prevent the device from using the default route.

`--bootproto=`

One of **`dhcp`**, **`bootp`**, **`ibft`**, or **`static`**. The default option is **`dhcp`**; the **`dhcp`** and **`bootp`** options are treated the same.

The DHCP method uses a DHCP server system to obtain its networking configuration. The BOOTP method is similar, requiring a BOOTP server to supply the networking configuration. To direct a system to use DHCP:

```
network --bootproto=dhcp
```

To direct a machine to use BOOTP to obtain its networking configuration, use the following line in the Kickstart file:

```
network --bootproto=bootp
```

To direct a machine to use the configuration specified in iBFT, use:

```
network --bootproto=ibft
```

The **`static`** method requires that you specify the IP address, netmask, gateway, and nameserver in the Kickstart file. This information is static and is used during and after the installation.

All static networking configuration information must be specified on *one* line; you cannot wrap lines using a backslash (\) as you can on a command line.

```
network --bootproto=static --ip=10.0.2.15 --netmask=255.255.255.0 --gateway=10.0.2.254 --nameserver=10.0.2.1
```

You can also configure multiple nameservers at the same time. To do so, use the **`--nameserver=`** options once for each name server you want to configure:

```
network --bootproto=static --ip=10.0.2.15 --netmask=255.255.255.0 --gateway=10.0.2.254 --nameserver=192.168.2.1 --nameserver=192.168.3.1
```

`--device=`

Specifies the device to be configured (and eventually activated in **Anaconda**) with the **`network`** command.

If the **`--device=`** option is missing on the *first* use of the **`network`** command, the value of the **`ksdevice=`** **Anaconda** boot option is used, if available. Note that this is considered deprecated behavior; in most cases, you should always specify a **`--device=`** for every **`network`** command.

The behavior of any subsequent **`network`** command in the same Kickstart file is unspecified if its **`--device=`** option is missing. Make sure you specify this option for any **`network`** command beyond the first.

You can specify a device to be activated in any of the following ways:

- the device name of the interface, for example, `em1`
- the MAC address of the interface, for example, `01:23:45:67:89:ab`
- the keyword **`link`**, which specifies the first interface with its link in the **up** state
- the keyword **`bootif`**, which uses the MAC address that **pxelinux** set in the *`BOOTIF`* variable. Set **`IPAPPEND 2`** in your **`pxelinux.cfg`** file to have **pxelinux** set the *`BOOTIF`* variable.

For example:

```
network --bootproto=dhcp --device=em1
```

`--ip=`
IP address of the device.

`--ipv6=`
IPv6 address of the device, in the form of *address*[/*prefix length*] - for example, **`3ffe:ffff:0:1::1/128`** . If *prefix* is omitted, **`64`** will be used. You can also use **`auto`** for automatic configuration, or **`dhcp`** for DHCPv6-only configuration (no router advertisements).

`--gateway=`
Default gateway as a single IPv4 address.

`--ipv6gateway=`
Default gateway as a single IPv6 address.

`--nodefroute`
Prevents the interface being set as the default route. Use this option when you activate additional devices with the **`--activate=`** option, for example, a NIC on a separate subnet for an iSCSI target.

`--nameserver=`
DNS name server, as an IP address. To specify more than one name server, use this parameter multiple times. For example:

```
network --bootproto=static --ip=10.0.2.15 --netmask=255.255.255.0 --gateway=10.0.2.254 --
nameserver=192.168.2.1 --nameserver=192.168.3.1
```

`--nodns`
Do not configure any DNS server.

`--netmask=`
Network mask for the installed system.

`--hostname=`
Host name for the installed system. The host name can be either a *fully-qualified domain name* (FQDN) in the format *hostname.domainname*, or a short host name with no domain name. Many networks have a *Dynamic Host Configuration Protocol* (`DHCP`) service that automatically supplies connected systems with a domain name; to allow the `DHCP` service to assign the domain name to this machine, only specify the short host name.

`--ethtool=`
Specifies additional low-level settings for the network device which will be passed to the **ethtool** program.

`--essid=`
The network ID for wireless networks.

`--wepkey=`
The WEP encryption key for wireless networks.

`--wpakey=`
The WPA encryption key for wireless networks.

`--onboot=`
Whether or not to enable the device at boot time.

`--dhcpclass=`
: The DHCP class.

`--mtu=`
: The MTU of the device.

`--noipv4`
: Disable IPv4 on this device.

`--noipv6`
: Disable IPv6 on this device.

`--bondslaves=`
: When this option is used, the network device specified in the **`--device=`** option will be created using slaves defined in the **`--bondslaves=`** option. For example:

```
network --device=mynetwork --bondslaves=em1,em2
```

The above command will create a bond device named **`mynetwork`** using the **`em1`** and **`em2`** interfaces as its slaves.

`--bondopts=`
: A list of optional parameters for a bonded interface, which is specified using the **`--bondslaves=`** and **`--device=`** options. Options in this list must be separated by commas ("**`,`**") or semicolons ("**`;`**"). If an option itself contains a comma, use a semicolon to separate the options. For example:

```
network --bondopts=mode=active-backup,balance-rr;primary=eth1
```

Available optional parameters are listed in the *Working with Kernel Modules* chapter of the *Fedora System Administrator's Guide*, available at *http://docs.fedoraproject.org/*.

Important

The **`--bondopts=mode=`** parameter only supports full mode names such as **`balance-rr`** or **`broadcast`**, not their numerical representations such as **`0`** or **`3`**.

`--vlanid=`
: Specifies virtual LAN (VLAN) ID number (802.1q tag) for the device created using the device specified in **`--device=`** as a parent. For example, **`network --device=em1 --vlanid=171`** will create a virtual LAN device **`em1.171`**.

`--interfacename=`
: Specify a custom interface name for a virtual LAN device. This option should be used when the default name generated by the **`--vlanid=`** option is not desirable. This option must be used along with **`--vlanid=`**. For example:

```
network --device=em1 --vlanid=171 --interfacename=vlan171
```

The above command will create a virtual LAN interface named **`vlan171`** on the **`em1`** device with an ID of **`171`**.

The interface name can be arbitrary (for example, **my-vlan**), but in specific cases, the following conventions must be followed:

- If the name contains a dot (**.**), it must take the form of ***NAME*.*ID***. The *NAME* is arbitrary, but the *ID* must be the VLAN ID. For example: **em1.171** or **my-vlan.171**.
- Names starting with **vlan** must take the form of **vlan*ID*** - for example, **vlan171**.

--teamslaves=

Team device specified by the **--device=** option will be created using slaves specified in this option. Slaves are separated by commas. A slave can be followed by its configuration, which is a single-quoted JSON string with double quotes escaped by the **** character. For example:

```
network --teamslaves="p3p1'{\"prio\": -10, \"sticky\": true}',p3p2'{\"prio\": 100}'"
```

See also the **--teamconfig=** option.

--teamconfig=

Double-quoted team device configuration which is a single-quoted JSON string with double quotes escaped by the **** character. The device name is specified by **--device=** option and its slaves and their configuration by **--teamslaves=** option. For example:

```

network --device team0 --activate --bootproto static --ip=10.34.102.222 --
netmask=255.255.255.0 --gateway=10.34.102.254 --nameserver=10.34.39.2 --
teamslaves="p3p1'{\"prio\": -10, \"sticky\": true}',p3p2'{\"prio\": 100}'" --
teamconfig="{\"runner\": {\"name\": \"activebackup\"}}"
```

--bridgeslaves=

When this option is used, the network bridge with device name specified using the **--device=** option will be created and devices defined in the **--bridgeslaves=** option will be added to the bridge. For example:

```
network --device=bridge0 --bridgeslaves=em1
```

--bridgeopts=

An optional comma-separated list of parameters for the bridged interface. Available values are **stp**, **priority**, **forward-delay**, **hello-time**, **max-age**, and **ageing-time**. For information about these parameters, see the *bridge setting* table in the `nm-settings(5)` man page or at *https://developer.gnome.org/NetworkManager/0.9/ref-settings.html*. Also see the *Fedora Networking Guide*, available at *http://docs.fedoraproject.org/*, for general information about network bridging.

A.4. Console and Environment

The following commands control the environment of the system after the installation finishes - language, keyboard layouts, or the graphical interface.

A.4.1. keyboard (optional) - Configure Keyboard Layouts

Sets one or more available keyboard layouts for the system.

```
keyboard --vckeymap= | --xlayouts= [--switch=]
```

--vckeymap=
: Specify a `VConsole` keymap which should be used. Valid names correspond to the list of files in the **/usr/lib/kbd/keymaps/*** directory, without the **.map.gz** extension.

--xlayouts=
: Specify a list of **X** layouts that should be used as a comma-separated list without spaces. Accepts values in the same format as `setxkbmap(1)`, either in the ***layout*** format (such as **cz**), or in the ***layout* (*variant*)** format (such as **cz (qwerty)**).

  All available layouts can be viewed on the `xkeyboard-config(7)` man page under **Layouts**.

--switch=
: Specify a list of layout-switching options (shortcuts for switching between multiple keyboard layouts). Multiple options must be separated by commas without spaces. Accepts values in the same format as `setxkbmap(1)`.

  Available switching options can be viewed on the `xkeyboard-config(7)` man page under **Options**.

The following example sets up two keyboard layouts (**English (US)** and **Czech (qwerty)**) using the **--xlayouts=** option, and allows to switch between them using **Alt**+**Shift**:

```
keyboard --xlayouts=us,'cz (qwerty)' --switch=grp:alt_shift_toggle
```

A.4.2. lang (optional) - Configure Language During Installation

Sets the language to use during installation and the default language to use on the installed system.

```
lang language [--addsupport=]
```

The file **/usr/share/system-config-language/locale-list** provides a list of the valid *language* codes in the first column of each line and is part of the *system-config-language* package.

Certain languages (for example, Chinese, Japanese, Korean, and Indic languages) are not supported during text-mode installation. If you specify one of these languages with the **lang** command and use text mode, the installation process will continue in English, but the installed system will use your selection as its default language.

--addsupport=
: Add support for additional languages. Takes the form of comma-separated list without spaces. For example:

```
lang en_US --addsupport=cs_CZ,de_DE,en_UK
```

A.4.3. services (optional) - Configure Services

Modifies the default set of services that will run under the default **systemd** target. The list of disabled services is processed before the list of enabled services - therefore, if a service appears on both lists, it will be enabled.

```
services [--disabled=list] [--enabled=list]
```

Do not include spaces in the list of services. If you do, Kickstart will enable or disable only the services up to the first space. For example:

```
services --disabled=auditd, cups,smartd, nfslock
```

The above will disable only the `auditd` service. To disable all four services, the entry should include no spaces:

```
services --disabled=auditd,cups,smartd,nfslock
```

`--disabled=`
: Disable the services given in the comma separated list.

`--enabled=`
: Enable the services given in the comma separated list.

A.4.4. skipx (optional) - Do Not Configure X Window System

If present, **X** will not be configured on the installed system.

> **Important**
>
> If you install a display manager among your package selection options, this package will create an **X** configuration, and the installed system will default to `graphical.target`. The effect of the **`skipx`** option will be overridden.

A.4.5. timezone (optional) - Configure Time Zone

Sets the system time zone to *`timezone`*. To view a list of available time zones, use the **`timedatectl list-timezones`** command.

```
timezone timezone [options]
```

`--utc`
: If present, the system assumes the hardware clock is set to UTC (Greenwich Mean) time.

`--nontp`
: Disable the NTP service automatic starting.

`--ntpservers=`
: Specify a list of NTP servers to be used as a comma-separated list without spaces.

A.4.6. xconfig (optional) - Configure X Window System

Configures the **X Window System**. If you install the **X Window System** with a Kickstart file that does not include the **`xconfig`** command, you must provide the **X** configuration manually during installation.

Do not use this command in a Kickstart file that does not install the **X Window System**.

`--defaultdesktop=`
: Specify either **`GNOME`** or **`KDE`** to set the default desktop (assumes that **GNOME Desktop Environment** and/or **KDE Desktop Environment** has been installed in the **`%packages`** section).

`--startxonboot`
: Use a graphical login on the installed system.

A.5. Users, Groups and Authentication

The commands below are used to control user accounts, groups, and related areas.

A.5.1. auth or authconfig (optional) - Configure Authentication

Sets up the authentication options for the system using the **`authconfig`** command, which can also be run on a command line after the installation finishes. See the `authconfig(8)` manual page and the **`authconfig --help`** command for more details. Passwords are shadowed by default.

```
auth [--enablenis | --nisdomain= | --nisserver= | --enableshadow | --enableldap | --
enableldapauth | --ldapserver= | --ldapbasedn= | --enableldaptls | --disableldaptls | --
enablekrb5 | --krb5realm= | --krb5kdc= | --krb5adminserver= | --enablehesiod | --hesiodlhs=
 | --hesiodrhs= | --enablesmbauth | --smbservers= | --smbworkgroup= | --enablecache | --
passalgo=]
```

`--enablenis`

Turns on NIS support. By default, **`--enablenis`** uses whatever domain it finds on the network. A domain should almost always be set by hand with the **`--nisdomain=`** option.

`--nisdomain=`

NIS domain name to use for NIS services.

`--nisserver=`

Server to use for NIS services (broadcasts by default).

`--useshadow`or **`--enableshadow`**

Use shadow passwords. Active by default.

`--enableldap`

Turns on LDAP support in **`/etc/nsswitch.conf`**, allowing your system to retrieve information about users (for example, their UIDs, home directories, and shells) from an LDAP directory. To use this option, you must install the *nss-pam-ldapd* package. You must also specify a server and a base *DN* (distinguished name) with **`--ldapserver=`** and **`--ldapbasedn=`**.

`--enableldapauth`

Use LDAP as an authentication method. This enables the `pam_ldap` module for authentication and changing passwords, using an LDAP directory. To use this option, you must have the *nss-pam-ldapd* package installed. You must also specify a server and a base DN with **`--ldapserver=`** and **`--ldapbasedn=`**. If your environment does not use *TLS* (Transport Layer Security), use the **`--disableldaptls`** switch to ensure that the resulting configuration file works.

`--ldapserver=`

If you specified either **`--enableldap`** or **`--enableldapauth`**, use this option to specify the name of the LDAP server to use. This option is set in the **`/etc/ldap.conf`** file.

`--ldapbasedn=`

If you specified either **`--enableldap`** or **`--enableldapauth`**, use this option to specify the DN in your LDAP directory tree under which user information is stored. This option is set in the **`/etc/ldap.conf`** file.

`--enableldaptls`

Use TLS (Transport Layer Security) lookups. This option allows LDAP to send encrypted usernames and passwords to an LDAP server before authentication.

`--disableldaptls`
Do not use TLS (Transport Layer Security) lookups in an environment that uses LDAP for authentication.

`--enablekrb5`
Use Kerberos 5 for authenticating users. Kerberos itself does not know about home directories, UIDs, or shells. If you enable Kerberos, you must make users' accounts known to this workstation by enabling LDAP, NIS, or Hesiod or by using the **`useradd`** command. If you use this option, you must have the *pam_krb5* package installed.

`--krb5realm=`
The Kerberos 5 realm to which your workstation belongs.

`--krb5kdc=`
The KDC (or KDCs) that serve requests for the realm. If you have multiple KDCs in your realm, use a comma-separated list without spaces.

`--krb5adminserver=`
The KDC in your realm that is also running kadmind. This server handles password changing and other administrative requests. This server must be run on the master KDC if you have more than one KDC.

`--enablehesiod`
Enables Hesiod support for looking up user home directories, UIDs, and shells. More information on setting up and using Hesiod on your network is in **`/usr/share/doc/glibc-2.x.x/README.hesiod`**, which is included in the *glibc* package. Hesiod is an extension of DNS that uses DNS records to store information about users, groups, and various other items.

`--hesiodlhs=` and **`--hesiodrhs=`**
The `Hesiod` LHS (left-hand side) and RHS (right-hand side) values, set in **`/etc/hesiod.conf`**. The `Hesiod` library uses these values to search DNS for a name, similar to the way that `LDAP` uses a base DN.

To look up user information for the username `jim`, the Hesiod library looks up **`jim.passwd`*`LHSRHS`***, which should resolve to a TXT record that contains a string identical to an entry for that user in the **`passwd`** file: **`jim:*:501:501:Jungle Jim:/home/jim:/bin/bash`**. To look up groups, the Hesiod library looks up **`jim.group`*`LHSRHS`*** instead.

To look up users and groups by number, make **`501.uid`** a CNAME for **`jim.passwd`**, and **`501.gid`** a CNAME for **`jim.group`**. Note that the library does not place a period (**`.`**) in front of the LHS and RHS values when performing a search. Therefore, if the LHS and RHS values need to have a period placed in front of them, you must include the period in the values you set for **`--hesiodlhs=`** and **`--hesiodrhs=`**.

`--enablesmbauth`
Enables authentication of users against an SMB server (typically a Samba or Windows server). SMB authentication support does not know about home directories, UIDs, or shells. If you enable SMB, you must make users' accounts known to the workstation by enabling LDAP, NIS, or Hesiod or by using the **`useradd`** command.

`--smbservers=`
The name of the servers to use for SMB authentication. To specify more than one server, separate the names with commas (**`,`**).

--smbworkgroup=
The name of the workgroup for the SMB servers.

--enablecache
Enables the `nscd` service. The `nscd` service caches information about users, groups, and various other types of information. Caching is especially helpful if you choose to distribute information about users and groups over your network using `NIS`, `LDAP`, or `Hesiod`.

--passalgo=
Specify **sha256** to set up the SHA-256 hashing algorithm or **sha512** to set up the SHA-512 hashing algorithm.

A.5.2. group (optional) - Create User Group

Creates a new user group on the system. If a group with the given name or GID already exists, this command will fail. In addition, the **user** command can be used to create a new group for the newly created user.

```
group --name=name [--gid=gid]
```

--name=
Provides the name of the group.

--gid=
The group ID (GID). If not provided, defaults to the next available non-system GID.

A.5.3. pwpolicy (optional) - Change the Default Password Policy

This command can be used to set custom requirements (policy) such as length and strength for passwords set during the installation - the `root` password, user passwords and LUKS (disk encryption) password.

```
pwpolicy name [--minlen=length] [--minquality=quality] [--strict|nostrict] [--emptyok|
noempty] [--changesok|nochanges]
```

The `libpwquality` library is used to check minimum password requirements (length and quality). You can use the **pwscore** and **pwmake** commands provided by the *libpwquality* package to check the quality score of your chosen password, or to create a random password with a given score. See the `pwscore(1)` and `pwmake(1)` man pages for details about these commands.

Important

This command must be used inside the **%anaconda** section. See *Section A.9, “%anaconda (optional) - Additional Anaconda Configuration”* for details.

name
Name of the password entry. Supported values are **root**, **user** and **luks** for `root` password, user passwords and LUKS password, respectively.

--minlen=
Sets the minimum allowed password length. The default minimum length is **8**.

`--minquality=`

Sets the minimum allowed password quality as defined by the `libpwquality` library. The default value is **`50`**.

`--strict`

Strict password enforcement. Passwords not meeting the quality requirements specified in **`--minquality=`** will not be allowed. Enabled by default.

`--notstrict`

Passwords *not* meeting the minimum quality requirements specified in **`--minquality=`** will be allowed after **Done** is clicked twice.

`--emptyok`

Allow empty passwords. Enabled by default.

`--notempty`

Do not allow empty passwords.

`--changesok`

Allow changing the password in the user interface, even if the Kickstart file already specifies a password.

`--nochanges`

Do not allow changing passwords which are already set in the Kickstart file. Enabled by default.

An example use of the **`pwpolicy`** command is below:

```
%anaconda
pwpolicy root --minlen=10 --minquality=60 --strict --notempty --nochanges
%end
```

A.5.4. realm (optional) - Join an Active Directory or IPA Domain

Join an Active Directory or IPA domain. For more information about this command, see the **`join`** section of the `realm(8)` man page.

```
realm join domain [options]
```

`--computer-ou=OU=`

Provide the *distinguished name* of an organizational unit in order to create the computer account. The exact format of the distinguished name depends on the client software and membership software. The root DSE portion of the distinguished name can usually be left out.

`--no-password`

Join automatically without a password.

`--one-time-password=`

Join using a one-time password. This is not possible with all types of realm.

`--client-software=`

Only join realms which can run this client software. Valid values include **`sssd`** and **`winbind`**. Not all realms support all values. By default, the client software is chosen automatically.

`--server-software=`
Only join realms which can run this server software. Possible values include **`active-directory`** or **`freeipa`**.

`--membership-software=`
Use this software when joining the realm. Valid values include **`samba`** and **`adcli`**. Not all realms support all values. By default, the membership software is chosen automatically.

A.5.5. rootpw (required) - Set Root Password

Sets the system's root password to the *`password`* argument.

```
rootpw [--iscrypted|--plaintext] [--lock] password
```

`--iscrypted`
If this option is present, the password argument is assumed to already be encrypted. This option is mutually exclusive with **`--plaintext`**. To create an encrypted password, you can use Python:

```
$ python -c 'import crypt; print(crypt.crypt("My Password", "$6$My Salt"))'
```

This will generate a SHA512 crypt of your password using your provided salt.

`--plaintext`
If this option is present, the password argument is assumed to be in plain text. This option is mutually exclusive with **`--iscrypted`**.

`--lock`
If this option is present, the root account is locked by default. This means that the root user will not be able to log in from the console.

A.5.6. selinux (optional) - Configure SELinux

Sets the state of SELinux on the installed system. The default policy is `enforcing`. For more information regarding SELinux in Fedora, see the *Fedora SELinux User's and Administrator's Guide*, available at *http://docs.fedoraproject.org/*.

```
selinux [--disabled|--enforcing|--permissive]
```

`--enforcing`
Enables SELinux with the default targeted policy being `enforcing`.

`--permissive`
Enables SELinux with the default targeted policy being `permissive`. This policy outputs warnings based on the SELinux policy, but does not actually enforce the policy.

`--disabled`
Disables SELinux completely.

A.5.7. sshkey (optional) - Add an Authorized SSH Key

Use this command to add a specified SSH key to an user's **`authorized_keys`** file. The specified user must either be `root`, or it must be created in the Kickstart file - either automatically by an installed package, or using *Section A.5.8, “user (optional) - Create User Account”*.

```
sshkey --username=user "ssh_key"
```

The *ssh_key* must be a full SSH key fingerprint, and it must be enclosed in quotes (`""`) because the key may contain spaces.

--username=
The name of the user account to which you want to install the SSH key.

A.5.8. user (optional) - Create User Account

Creates a new user on the system.

```
user --name=username [options]
```

--name=
Provides the name of the user. This option is required.

--gecos=
Provides the GECOS information for the user. This is a string of various system-specific fields separated by a comma. It is frequently used to specify the user's full name, office number, etc. See the `passwd(5)` man page for more details.

--groups=
In addition to the default group, a comma separated list of group names the user should belong to. The groups must exist before the user account is created. See *Section A.5.2, “group (optional) - Create User Group”*.

--homedir=
The home directory for the user. If not provided, this defaults to **/home/*username***.

--lock
If this option is present, this account is locked by default. This means that the user will not be able to log in from the console.

--password=
The new user's password. If no password is provided, the account will be locked.

--password=
The new user's password. If not provided, the account will be locked by default.

--iscrypted
If this option is present, the password argument is assumed to already be encrypted. This option is mutually exclusive with **--plaintext**. To create an encrypted password, you can use Python:

```
$ python -c 'import crypt; print(crypt.crypt("My Password", "$6$My Salt"))'
```

This will generate a SHA512 crypt of your password using your provided salt.

--plaintext
If this option is present, the password argument is assumed to be in plain text. This option is mutually exclusive with **--iscrypted**.

--shell=
The user's login shell. If not provided, the system default will be used.

`--uid=`

The *UID* (User ID). If not provided, this defaults to the next available non-system UID.

`--gid=`

The *GID* (Group ID) to be used for the user's default group. If not provided, this defaults to the next available non-system group ID.

A.6. Installation Environment

The following commands control how the system will behave during the installation.

A.6.1. autostep (optional) - Go Through Every Screen

Normally, Kickstart installations skip unnecessary screens. This option makes the installation program step through every screen, displaying each briefly. This option should not be used when deploying a system because it may disrupt package installation.

```
autostep [--autoscreenshot]
```

`--autoscreenshot`

Take a screenshot at every step during installation. These screenshots are stored in **`/tmp/anaconda-screenshots`** during the installation, and after the installation finishes you can find them in **`/root/anaconda-screenshots`**.

Each screen is only captured right before the installer switches to the next one. This is important, because if you do not use all required Kickstart options and the installation therefore does not begin automatically, you can go to the screens which were not automatically configured, perform any configuration you want. Then, when you press **Done** to continue, the screen will be captured including the configuration you just provided.

A.6.2. cmdline (optional) - Perform Installation in Command Line Mode

Perform the installation in a completely non-interactive command line mode. Any prompts for interaction halts the install. This mode is useful on IBM System z systems with the x3270 terminal.

A.6.3. graphical (optional) - Perform Installation in Graphical Mode

Perform the installation in graphical mode. This is the default. This command takes no options.

A.6.4. logging (optional) - Configure Error Logging During Installation

Controls the error logging of **Anaconda** during installation. It has no effect on the installed system.

```
logging [--host= | --port= | --level=]
```

`--host=`

Send logging information to the given remote host, which must be running a `syslogd` process configured to accept remote logging.

`--port=`

If the remote syslogd process uses a port other than the default, it may be specified with this option.

`--level=`
Specify the minimum level of messages that appear on virtual console 3 (`tty3`). This only affects messages printed to the console; log files will contain messages of all levels. Possible values are **`debug`**, **`info`**, **`warning`**, **`error`**, or **`critical`**.

A.6.5. rescue (optional) - Rescue Mode

Automatically enters the installation program's rescue mode. This gives you a chance to repair the system in case of any problems.

```
rescue [--nomount|--romount]
```

`--nomount` or **`--romount`**
Controls how the installed system is mounted in the rescue environment. By default, the installation program will find your system and mount it in read-write mode, telling you where it has performed this mount. You may optionally choose to not mount anything (the **`--nomount`** option) or mount in read-only mode (the **`--romount`** option). Only one of these two options may be used.

A.6.6. sshpw (optional) - Restrict ssh Access During Installation

During the installation, you can interact with the installation program and monitor its progress over an SSH connection. Use the **`sshpw`** command to create temporary accounts through which to log on. Each instance of the command creates a separate account that exists only in the installation environment. These accounts are not transferred to the installed system.

```
sshpw --username=name password [--iscrypted|--plaintext] [--lock]
```

Important

By default, the `ssh` server is not started during the installation. To make `ssh` available during the installation, boot the system with the kernel boot option **`inst.sshd`**. See *Section 8.2.3, “Console, Environment and Display Options”* for details.

Note

If you want to disable root `ssh` access to your hardware during installation, use the following:

```
sshpw --username=root --lock
```

`--username`
Provides the name of the user. This option is required.

`--iscrypted`
If this option is present, the password argument is assumed to already be encrypted. This option is mutually exclusive with **`--plaintext`**. To create an encrypted password, you can use Python:

```
$ python -c 'import crypt; print(crypt.crypt("My Password", "$6$My Salt"))'
```

This will generate a SHA512 crypt of your password using your provided salt.

`--plaintext`
If this option is present, the password argument is assumed to be in plain text. This option is mutually exclusive with **`--iscrypted`**

`--lock`
If this option is present, this account is locked by default. This means that the user will not be able to log in from the console.

A.6.7. text (optional) - Perform Installation in Text Mode

Perform the Kickstart installation in text mode. Kickstart installations are performed in graphical mode by default.

A.6.8. unsupported_hardware (optional) - Suppress Unsupported Hardware Alerts

Suppress the **Unsupported Hardware Detected** alert. If this command is not included and unsupported hardware is detected, the installation will stall at this alert.

A.6.9. vnc (optional) - Configure VNC Access

Allows the graphical installation to be viewed remotely via VNC. This method is usually preferred over text mode, as there are some size and language limitations in text installations. With no additional options, this command will start a VNC server on the installation system with no password and will display the details required to connect to it.

```
vnc [--host=hostname] [--port=port] [--password=password]
```

For more information about VNC installations, including instructions on how to connect to the installation system, see *Chapter 11, Installing Using VNC*.

`--host=`
Connect to a VNC viewer listening on the given hostname.

`--port=`
Provide a port that the remote VNC viewer process is listening on. If not provided, the VNC default (**`5900`**) will be used.

`--password=`
Set a password which must be provided to connect to the VNC session. This is optional, but recommended.

A.7. After the Installation

This section contains commands which control the system's behavior immediately after the installation finishes.

A.7.1. %addon com_redhat_kdump (optional) - Configure kdump

This command configures the **kdump** kernel crash dumping mechanism.

> **Note**
>
> The syntax for this command is unusual because it is an add-on rather than a built-in Kickstart command. For more information about add-ons, see *Section A.8, “%addon (optional) - Include an Anaconda Add-on”*.

Kdump is a kernel crash dumping mechanism that allows you to save the contents of the system's memory for later analysis. It relies on **kexec**, which can be used to start a Linux kernel from the context of another kernel without rebooting the system and preserve the contents of the first kernel memory that would otherwise be lost.

In case of a system crash, **kexec** starts a second kernel (a *capture kernel*). This capture kernel resides in a reserved part of the system memory that is inaccessible to the main kernel. **Kdump** then captures the contents of the crashed kernel's memory (a *crash dump*) and saves it to a specified location. This location can not be configured using Kickstart; it must be specified after the installation by editing the **`/etc/kdump.conf`** configuration file.

Available options are:

`--enable`
Enable **Kdump** on the installed system.

`--disable`
Do not enable **Kdump** on the installed system.

`--reserve-mb=`
The amount of memory you want to reserve for **Kdump** in megabytes. For example:

```
%addon com_redhat_kdump --enable --reserve-mb=128
%end
```

You can also specify **`auto`** instead of a numeric value. In that case the installer will determine the amount of RAM to reserve for kdump automatically, based on your system architecture and the total amount of memory on the system.

If you enable **Kdump** and do not specify the **`--reserve-mb=`** option, the **`auto`** value will be used.

`--enablefadump`
Enable firmware-assisted dumping on systems which allow it (notably IBM Power Systems servers).

A.7.2. firstboot (optional) - Enable or Disable Initial Setup

Determine whether the **Initial Setup** application starts the first time the system is booted. If enabled, the *initial-setup* package must be installed. If not specified, this option is disabled by default. For more information about **Initial Setup**, see *Section 6.1, “Initial Setup”*.

```
firstboot --enable|--disable [--reconfig]
```

`--enable` or **`--enabled`**
: **Initial Setup** will be started the first time the installed system boots.

`--disable` or **`--disabled`**
: **Initial Setup** will be disabled.

`--reconfig`
: **Initial Setup** will start after the reboot in reconfiguration mode. This mode enables the language, mouse, keyboard, root password, security level, time zone and networking configuration options in addition to the default ones.

A.7.3. halt (optional) - Halt System After Installation

Halt the system after the installation has successfully completed. This is similar to a manual installation, where after the installation finishes, the installer displays a message and waits for the user to press a key before rebooting. During a Kickstart installation, if no completion method is specified, this option is used as the default.

For other completion methods, see the **`poweroff`**, **`reboot`**, and **`shutdown`** commands.

A.7.4. poweroff (optional) - Power Off After Installation

Shut down and power off the system after the installation has successfully completed.

Note

The **`poweroff`** command is highly dependent on the system hardware in use. Specifically, certain hardware components such as the BIOS, APM (advanced power management), and ACPI (advanced configuration and power interface) must be able to interact with the system kernel. Consult your hardware documentation for more information on you system's APM/ACPI abilities.

For other completion methods, see the **`halt`**, **`reboot`**, and **`shutdown`** Kickstart commands.

A.7.5. reboot (optional) - Reboot After Installation

Reboot after the installation is successfully completed. If you are installing Fedora on IBM System z in command line mode (using *Section A.6.2, “cmdline (optional) - Perform Installation in Command Line Mode”*), this command is necessary for a fully automated installation.

For other completion methods, see the **`halt`**, **`poweroff`**, and **`shutdown`** Kickstart options.

Important

Use of the **`reboot`** command *may* result in an endless installation loop, depending on the installation media and method.

`--eject`
: Attempt to eject the installation media (if installing from a DVD) before rebooting.

A.7.6. shutdown (optional) - Shut Down After Installation

Shut down the system after the installation has successfully completed.

For other completion methods, see the **`halt`**, **`poweroff`**, and **`reboot`** Kickstart options.

A.8. %addon (optional) - Include an Anaconda Add-on

You can expand the basic **Anaconda** and Kickstart functionality by using custom add-ons specified using the **`%addon`** section.

To use an add-on in your Kickstart file, add the **`%addon`** ***`addon_name options`*** command. The section must be closed with an **`%end`** statement. For example, to use the **Kdump** add-on, which is included by default on some installations, use the following block:

```
%addon com_redhat_kdump --enable --reserve-mb=128%end
```

The **`%addon`** section does not have any options of its own; all options depend on the add-on being used.

For more information about **Anaconda** add-ons, see the *Fedora Anaconda Addon Development Guide*[1].

A.9. %anaconda (optional) - Additional Anaconda Configuration

This section, introduced in Fedora 22, is used to specify additional configuration options for the installer. Commands inside this section only control the behavior during installation, not on the installed system. The only command currently available inside this section is *Section A.5.3, "pwpolicy (optional) - Change the Default Password Policy"*. The **`%anaconda`** section must end with **`%end`**.

In interactive installations (using the graphical or text interface), the **`/usr/share/anaconda/interactive-defaults.ks`** file contains the default **`%anaconda`** section. To change the defaults, you must create a **`product.img`** file with a Kickstart file replacing the default one, and pass this file to **Anaconda** using a boot option.

When using a Kickstart file, you can override the default **`%anaconda`** section by using the section again in your custom Kickstart.

A.10. %include (optional) - Include Contents of Another File

Use the **`%include`** ***`/path/to/file`*** command to include the contents of another file in the Kickstart file as though the contents were at the location of the **`%include`** command in the Kickstart file.

A.11. %ksappend (optional) - Append Contents of Another File

The **`%ksappend`** ***`url`*** directive is very similar to *Section A.10, "%include (optional) - Include Contents of Another File"* in that it is used to include the contents of additional files as though they were at the location of the **`%ksappend`** command. The difference is in when the two directives are processed.

[1] https://vpodzime.fedorapeople.org/anaconda-addon-development-guide/

%ksappend is processed in an initial pass, before any other part of the Kickstart file. Then, this expanded Kickstart file is passed to the rest of **Anaconda** where all **%pre** scripts are handled, and then finally the rest of the Kickstart file is processed in order, which includes **%include** directives.

Therefore, **%ksappend** provides a way to include a file containing **%pre** scripts, while **%include** does not.

A.12. %packages (required) - Package Selection

Use the **%packages** command to begin a Kickstart section which describes the software packages to be installed. This section must end with an **%end** statement.

You can specify packages by *environment*, *group*, or by their package names. Several environments and groups that contain related packages are defined. See the **repodata/comps.xml** file in your installation source for environment and group definitions.

The **comps.xml** file contains a structure describing available environments (marked by the **<environment>** tag) and groups (the **<group>** tag). Each entry has an ID, user visibility value, name, description, and package list. If the group is selected for installation, the packages marked **mandatory** in the package list are always installed, the packages marked **default** are installed if they are not specifically excluded, and the packages marked **optional** must be specifically included even when the group is selected.

You can specify a package group or environment using either its ID (the **<id>** tag) or name (the **<name>** tag).

Important

To install a 32-bit package on a 64-bit system, you will need to append the package name with the 32-bit architecture for which the package was built - for example, *glibc.i686*. The **--multilib** option also must be specified in the Kickstart file; see the available options below.

Important

Initial Setup does not run after a system is installed from a Kickstart file unless a desktop environment and the **X Window System** were included in the installation and graphical login was enabled. This means that by default, no users except for root will be created. You can either create a user with the **user** option in the Kickstart file before installing additional systems from it (see *Section A.5.8, “user (optional) - Create User Account”* for details) or log into the installed system with a virtual console as root and add users with the **useradd** command.

Specifying Environments, Groups and Packages

Specifying an Environment

In addition to groups, you specify an entire environment to be installed:

```
%packages
@^Infrastructure Server
%end
```

This command will install all packages which are part of the **Infrastructure Server** environment. All available environments are described in the **comps.xml** file.

Specifying Groups

Specify groups, one entry to a line, starting with an @ symbol, and then the full group name or group id as given in the **comps.xml** file. For example:

```
%packages
@X Window System
@Desktop
@Sound and Video
%end
```

The **Core** and **Base** groups are always selected - it is not necessary to specify them in the **%packages** section.

The **comps.xml** file also defines groups called **Conflicts (*variant*)** for each variant of Fedora. This group contains all packages which are known to cause file conflicts, and is intended to be excluded.

Specifying Individual Packages

Specify individual packages by name, one entry to a line. You can use the asterisk character (*****) as a *wildcard* in package names. For example:

```
%packages
sqlite
curl
aspell
docbook*
%end
```

The **docbook*** entry includes the packages *docbook-dtds*, *docbook-simple*, *docbook-slides* and others that match the pattern represented with the wildcard.

Excluding Environments, Groups, or Packages

Use a leading dash (**-**) to specify packages or groups to exclude from the installation. For example:

```
%packages
-@Graphical Internet
-autofs
-ipa*fonts
%end
```

Important

Installing all available packages using only ***** in a Kickstart file is not supported, even if you exclude the **@Conflicts (*variant*)** group.

You can change the default behavior of the **%packages** section by using several options. Some options work for the entire package selection, others are used with only specific groups.

Common Package Selection Options

The following options are available for the **%packages**. To use an option, append it to the start of the package selection section. For example:

```
%packages --multilib --ignoremissing
```

--nocore

Do not install the **@Core** group.

--ignoremissing

Ignore any packages, groups and environments missing in the installation source, instead of halting the installation to ask if the installation should be aborted or continued.

--excludedocs

Do not install any documentation contained within packages. In most cases, this will exclude any files normally installed in the **/usr/share/doc*** directory, but the specific files to be excluded depend on individual packages.

--multilib

Configure the installed system for multilib packages (that is, to allow installing 32-bit packages on a 64-bit system) and install packages specified in this section as such.

Normally, on a 64-bit system, only packages for this architecture (marked as **x86_64**) and packages for all architectures (marked as **noarch**) would be installed. When you use this option, packages for 32-bit systems (marked as **i686**) will be automatically installed as well, if available.

This only applies to packages explicitly specified in the **%packages** section. Packages which are only being installed as dependencies without being specified in the Kickstart file will only be installed in architecture versions in which they are needed, even if they are available for more architectures.

Options for Specific Package Groups

The options in this list only apply to a single package group. Instead of using them at the **%packages** command in the Kickstart file, append them to the group name. For example:

```
%packages
@Graphical Internet --optional
%end
```

--nodefaults

Only install the group's mandatory packages, not the default selections.

--optional

Install packages marked as optional in the group definition in the **comps.xml** file, in addition to installing the default selections.

Note that some package groups, such as **Scientific Support**, do not have any mandatory or default packages specified - only optional packages. In this case the **--optional** option must always be used, otherwise no packages from this group will be installed.

A.13. %pre (optional) - Pre-installation Script

You can add commands to run on the system immediately after the Kickstart file has been parsed, but before the installation begins. This section must be placed towards the end of the Kickstart file, after the actual Kickstart commands, and must start with **`%pre`** and end with **`%end`**. If your Kickstart file also includes a **`%post`** section, the order in which the **`%pre`** and **`%post`** sections are included does not matter.

You can access the network in the **`%pre`** section. However, the *name service* has not been configured at this point, so only IP addresses work, not URLs.

The pre-installation script section of Kickstart *cannot* manage multiple install trees or source media. This information must be included for each created Kickstart file, as the pre-installation script occurs during the second stage of the installation process.

> **Note**
>
> Unlike the post-installation script, the pre-installation script is not run in the `chroot` environment.

The following options can be used to change the behavior of pre-installation scripts. To use an option, append it to the **`%pre`** line at the beginning of the script. For example:

```
%pre --interpreter=/usr/bin/python
--- Python script omitted --
%end
```

`--interpreter=`

Allows you to specify a different scripting language, such as Python. Any scripting language available on the system can be used; in most cases, these will be `/usr/bin/sh`, `/usr/bin/bash`, and `/usr/bin/python`.

`--erroronfail`

Display an error and halt the installation if the script fails. The error message will direct you to where the cause of the failure is logged.

`--log=`

Logs the script's output into the specified log file. For example:

```
%pre --log=/mnt/sysimage/root/ks-pre.log
```

For an example of a pre-installation script, see *Section A.15.2, “Example Pre-installation Script”*.

A.14. %post (optional) - Post-installation Script

You have the option of adding commands to run on the system once the installation is complete, but before the system is rebooted for the first time. This section must be placed towards the end of the Kickstart file, after the actual Kickstart commands, and must start with **`%post`** and end with **`%end`**. If your Kickstart file also includes a **`%pre`** section, the order of the **`%pre`** and **`%post`** sections does not matter.

This section is useful for functions such as installing additional software or configuring an additional name server. The post-install script is run in a chroot environment, therefore, performing tasks such as copying scripts or RPM packages from the installation media do not work by default. You can change this behavior using the **--nochroot** option as described below.

> **Important**
>
> If you configured the network with static IP information, including a name server, you can access the network and resolve IP addresses in the **%post** section. If you configured the network for DHCP, the **/etc/resolv.conf** file has not been completed when the installation executes the **%post** section. You can access the network, but you cannot resolve IP addresses. Thus, if you are using DHCP, you must specify IP addresses in the **%post** section.

The following options can be used to change the behavior of post-installation scripts. To use an option, append it to the **%post** line at the beginning of the script. For example:

```
%post --interpreter=/usr/bin/python
--- Python script omitted --
%end
```

--interpreter=

Allows you to specify a different scripting language, such as Python. For example:

```
%post --interpreter=/usr/bin/python
```

Any scripting language available on the system can be used; in most cases, these will be /usr/bin/sh, /usr/bin/bash, and /usr/bin/python.

--nochroot

Allows you to specify commands that you would like to run outside of the chroot environment.

The following example copies the file **/etc/resolv.conf** to the file system that was just installed.

```
%post --nochroot
cp /etc/resolv.conf /mnt/sysimage/etc/resolv.conf
%end
```

--erroronfail

Display an error and halt the installation if the script fails. The error message will direct you to where the cause of the failure is logged.

--log=

Logs the script's output into the specified log file. Note that the path of the log file must take into account whether or not you use the **--nochroot** option. For example, without **--nochroot**:

```
%post --log=/root/ks-post.log
```

with **--nochroot**:

```
%post --nochroot --log=/mnt/sysimage/root/ks-post.log
```

For an example of a post-installation script, see *Section A.15.3, “Example Post-installation Script”*.

A.15. Example Kickstart Configurations

A.15.1. Advanced Partitioning Example

The following is an integrated example showing the **`clearpart`**, **`zerombr`**, **`part`**, **`raid`**, **`volgroup`**, and **`logvol`** Kickstart options in action:

Example A.3. Advanced Partitioning Example

```

clearpart --drives=hda,hdc
zerombr
# Raid 1 IDE config
part raid.11 --size 1000 --asprimary --ondrive=hda
part raid.12 --size 1000 --asprimary --ondrive=hda
part raid.13 --size 2000 --asprimary --ondrive=hda
part raid.14 --size 8000 --ondrive=hda
part raid.15 --size 16384 --grow --ondrive=hda
part raid.21 --size 1000 --asprimary --ondrive=hdc
part raid.22 --size 1000 --asprimary --ondrive=hdc
part raid.23 --size 2000 --asprimary --ondrive=hdc
part raid.24 --size 8000 --ondrive=hdc
part raid.25 --size 16384 --grow --ondrive=hdc

# You can add --spares=x
raid / --fstype xfs --device root --level=RAID1 raid.11 raid.21
raid /safe --fstype xfs --device safe --level=RAID1 raid.12 raid.22
raid swap --fstype swap --device swap --level=RAID1 raid.13 raid.23
raid /usr --fstype xfs --device usr --level=RAID1 raid.14 raid.24
raid pv.01 --fstype xfs --device pv.01 --level=RAID1 raid.15 raid.25

# LVM configuration so that we can resize /var and /usr/local later
volgroup sysvg pv.01
logvol /var --vgname=sysvg --size=8000 --name=var
logvol /var/freespace --vgname=sysvg --size=8000 --name=freespacetouse
logvol /usr/local --vgname=sysvg --size=1 --grow --name=usrlocal
```

This advanced example implements LVM over RAID, as well as the ability to resize various directories for future growth.

First, the **`clearpart`** command is used on drives `hda` and `hdc` to wipe them. The **`zerombr`** command initializes unused partition tables.

Then, the two drives are partitioned to prepare them for RAID configuration. Each drive is divided into five partitions, and each drive is partitioned into an identical layout.

The next part uses these pairs of physical partitions to create a software RAID device with RAID1 level (mirroring). The first four RAID devices are used for **`/`** (root), **`/safe`**, swap and **`/usr`**. The fifth, largest pair of partitions is named **`pv.01`** and will be used in the following part as a physical volume for LVM.

Finally, the last set of commands first creates a volume group named **`sysvg`** on the **`pv.01`** physical volume. Then, three logical volumes (**`/var`**, **`/var/freespace`** and **`/usr/local`**) are created and added to the **sysvg** volume group. The **/var** and **/var/freespace** volumes have a set size of 8 GB, and the **/usr/local** volume uses the **--grow** option to fill all remaining available space.

Important

The above example uses identifiers hda and hdc to identify disk drives. You should use unique identifiers, such as a disk labels or an UUIDs, to identify disk drives. See the note in introduction to this appendix.

A.15.2. Example Pre-installation Script

The following is an example **%pre** section:

Example A.4. Sample %pre Script

```
%pre
#!/bin/sh
hds=""
mymedia=""
for file in /proc/ide/h* do
mymedia=`cat $file/media`
if [ $mymedia == "disk" ] ; then
hds="$hds `basename $file`"
fi
done
set $hds
numhd=`echo $#`
drive1=`echo $hds | cut -d' ' -f1`
drive2=`echo $hds | cut -d' ' -f2`

#Write out partition scheme based on whether there are 1 or 2 hard drives
if [ $numhd == "2" ] ; then
#2 drives
echo "#partitioning scheme generated in %pre for 2 drives" > /tmp/part-include
echo "clearpart --all" >> /tmp/part-include
echo "part /boot --fstype xfs --size 75 --ondisk hda" >> /tmp/part-include
echo "part / --fstype xfs --size 1 --grow --ondisk hda" >> /tmp/part-include
echo "part swap --recommended --ondisk $drive1" >> /tmp/part-include
echo "part /home --fstype xfs --size 1 --grow --ondisk hdb" >> /tmp/part-include
else
#1 drive
echo "#partitioning scheme generated in %pre for 1 drive" > /tmp/part-include
echo "clearpart --all" >> /tmp/part-include
echo "part /boot --fstype xfs --size 75" >> /tmp/part-include
echo "part swap --recommended" >> /tmp/part-include
echo "part / --fstype xfs --size 2048" >> /tmp/part-include
echo "part /home --fstype xfs --size 2048 --grow" >> /tmp/part-include
fi
%end
```

This script determines the number of hard drives in the system and writes a text file with a different partitioning scheme depending on whether it has one or two drives. Instead of having a set of partitioning commands in the Kickstart file, include the following line:

```
%include /tmp/part-include
```

The partitioning commands selected in the script will be used.

A.15.3. Example Post-installation Script

The following is an example **%post** section:

Example A.5. Sample %post Script

```
# Start of the %post section with logging into /root/ks-post.log
%post --log=/root/ks-post.log

# Mount an NFS share
mkdir /mnt/temp
mount -o nolock 10.10.0.2:/usr/new-machines /mnt/temp
openvt -s -w -- /mnt/temp/runme
umount /mnt/temp

# End of the %post section
%end
```

The above example mounts an NFS share and executes a script named **runme** located at **/usr/new-machines/** on the share. Note that NFS file locking is *not* supported while in Kickstart mode, therefore the **-o nolock** option is required.

Appendix B. An Introduction to Disk Partitions

Note

This appendix is not necessarily applicable to architectures other than AMD64 and Intel 64. However, the general concepts mentioned here may apply.

This section discusses basic disk concepts, disk repartitioning strategies, the partition naming scheme used by Linux systems, and related topics.

If you are comfortable with disk partitions, you can skip ahead to *Section B.2, “Strategies for Disk Repartitioning”* for more information on the process of freeing up disk space to prepare for a Fedora installation.

B.1. Hard Disk Basic Concepts

Hard disks perform a very simple function - they store data and reliably retrieve it on command.

When discussing issues such as disk partitioning, it is important to have a understanding of the underlying hardware; however, since the theory is very complicated and expansive, only the basic concepts will be explained here. This appendix uses a set of simplified diagrams of a disk drive to help explain what is the process and theory behind partitions.

Figure B.1, “An Unused Disk Drive”, shows a brand-new, unused disk drive.

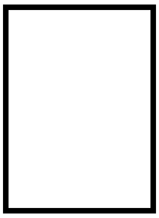


Figure B.1. An Unused Disk Drive

B.1.1. File Systems

To store data on a disk drive, it is necessary to *format* the disk drive first. Formatting (usually known as "making a *file system*") writes information to the drive, creating order out of the empty space in an unformatted drive.

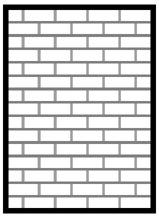


Figure B.2. Disk Drive with a File System

As *Figure B.2, “Disk Drive with a File System”*, implies, the order imposed by a file system involves some trade-offs:

- A small percentage of the driver's available space is used to store file system-related data and can be considered as overhead.
- A file system splits the remaining space into small, consistently-sized segments. For Linux, these segments are known as *blocks*. [1]

Note that there is no single, universal file system. As *Figure B.3, “Disk Drive with a Different File System”*, shows, a disk drive may have one of many different file systems written on it. Different file systems tend to be incompatible; that is, an operating system that supports one file system (or a handful of related file system types) may not support another. However, Fedora supports a wide variety of file systems (including many commonly used by other operating systems such as Microsoft Windows), making data interchange between different file systems easy.

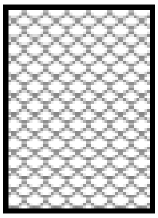


Figure B.3. Disk Drive with a Different File System

Writing a file system to disk is only the first step. The goal of this process is to actually *store* and *retrieve* data. The figure below shows a drive disk after some data have been written to it:

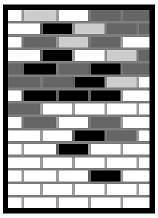


Figure B.4. Disk Drive with Data Written to It

As *Figure B.4, “Disk Drive with Data Written to It”*, shows, some of the previously-empty blocks are now holding data. However, by just looking at this picture, we cannot determine exactly how many files reside on this drive. There may only be one file or many, as all files use at least one block and some files use multiple blocks. Another important point to note is that the used blocks do not have to form a contiguous region; used and unused blocks may be interspersed. This is known as *fragmentation*. Fragmentation can play a part when attempting to resize an existing partition.

As with most computer-related technologies, disk drives changed over time after their introduction. In particular, they got bigger. Not larger in physical size, but bigger in their capacity to store information. And, this additional capacity drove a fundamental change in the way disk drives were used.

B.1.2. Partitions: Turning One Drive Into Many

Disk drives can be divided into *partitions*. Each partition can be accessed as if it was a separate disk. This is done through the addition of a *partition table*.

[1] Blocks really *are* consistently sized, unlike our illustrations. Keep in mind, also, that an average disk drive contains thousands of blocks. The picture is simplified for the purposes of this discussion.

There are several reasons for allocating disk space into separate disk partitions, for example:

- Logical separation of the operating system data from the user data
- Ability to use different file systems
- Ability to run multiple operating systems on one machine

There are currently two partitioning layout standards for physical hard disks: Master Boot Record (MBR) and GUID Partition Table (GPT). MBR is an older method of disk partitioning used with BIOS-based computers. GPT is a newer partitioning layout that is a part of the Unified Extensible Firmware Interface (UEFI). This section and *Section B.1.3, “Partitions Within Partitions - An Overview of Extended Partitions”* mainly describe the *Master Boot Record* (MBR) disk partitioning scheme. For information about the *GUID Partition Table* (GPT) partitioning layout, see *Section B.1.4, “GUID Partition Table (GPT)”*.

Note

While the diagrams in this chapter show the partition table as being separate from the actual disk drive, this is not entirely accurate. In reality, the partition table is stored at the very start of the disk, before any file system or user data. But for clarity, they are separate in our diagrams.

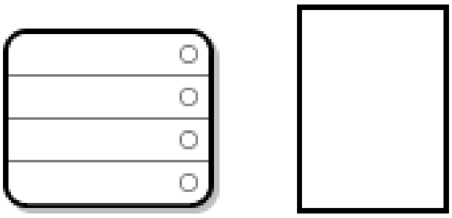


Figure B.5. Disk Drive with Partition Table

As *Figure B.5, “Disk Drive with Partition Table”* shows, the partition table is divided into four sections or four *primary* partitions. A primary partition is a partition on a hard drive that can contain only one logical drive (or section). Each section can hold the information necessary to define a single partition, meaning that the partition table can define no more than four partitions.

Each partition table entry contains several important characteristics of the partition:

- The points on the disk where the partition starts and ends
- Whether the partition is "active"
- The partition's type

The starting and ending points define the partition's size and location on the disk. The "active" flag is used by some operating systems' boot loaders. In other words, the operating system in the partition that is marked "active" is booted.

The type is a number that identifies the partition's anticipated usage. Some operating systems use the partition type to denote a specific file system type, to flag the partition as being associated with a particular operating system, to indicate that the partition contains a bootable operating system, or some combination of the three.

See *Figure B.6, “Disk Drive With Single Partition”* for an example of a disk drive with single partition.

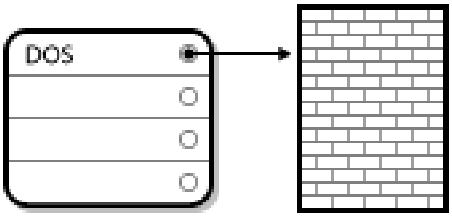


Figure B.6. Disk Drive With Single Partition

The single partition in this example is labeled as **DOS**. This label shows the *partition type*, with **DOS** being one of the most common ones. The table below shows a list of some of the commonly used partition types and hexadecimal numbers used to represent them.

Table B.1. Partition Types

Partition Type	Value	Partition Type	Value
Empty	00	Novell Netware 386	65
DOS 12-bit FAT	01	PIC/IX	75
XENIX root	02	Old MINIX	80
XENIX usr	03	Linux/MINUX	81
DOS 16-bit <=32M	04	Linux swap	82
Extended	05	Linux native	83
DOS 16-bit >=32	06	Linux extended	85
OS/2 HPFS	07	Amoeba	93
AIX	08	Amoeba BBT	94
AIX bootable	09	BSD/386	a5
OS/2 Boot Manager	0a	OpenBSD	a6
Win95 FAT32	0b	NEXTSTEP	a7
Win95 FAT32 (LBA)	0c	BSDI fs	b7
Win95 FAT16 (LBA)	0e	BSDI swap	b8
Win95 Extended (LBA)	0f	Syrinx	c7
Venix 80286	40	CP/M	db
Novell	51	DOS access	e1
PReP Boot	41	DOS R/O	e3
GNU HURD	63	DOS secondary	f2
Novell Netware 286	64	BBT	ff

B.1.3. Partitions Within Partitions - An Overview of Extended Partitions

In case four partitions are insufficient for your needs, you can use *extended partitions* to create up additional partitions. You do this by setting the type of a partition to "Extended".

An extended partition is like a disk drive in its own right - it has its own partition table which points to one or more partitions (now called *logical partitions*, as opposed to the four *primary partitions*) contained entirely within the extended partition itself. *Figure B.7, “Disk Drive With Extended Partition”*, shows a disk drive with one primary partition and one extended partition containing two logical partitions (along with some unpartitioned free space).

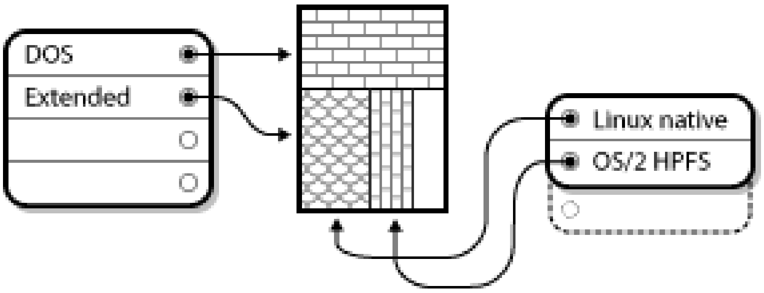


Figure B.7. Disk Drive With Extended Partition

As this figure implies, there is a difference between primary and logical partitions - there can only be four primary partitions, but there is no fixed limit to the number of logical partitions that can exist. However, due to the way in which partitions are accessed in Linux, no more than 12 logical partitions should be defined on a single disk drive.

B.1.4. GUID Partition Table (GPT)

GUID Partition Table (GPT) is a newer partitioning scheme based on using Globally Unique Identifiers (GUID). GPT was developed to cope with limitations of the MBR partition table, especially with the limited maximum addressable storage space of a disk. Unlike MBR, which is unable to address storage space larger than 2.2 terabytes, GPT can be used with hard disks larger than this; the maximum addressable disk size is 2.2 zettabytes. In addition, GPT by default supports creating up to 128 primary partitions. This number could be extended by allocating more space to the partition table.

GPT disks use logical block addressing (LBA) and the partition layout is as follows:

- To preserve backward compatibility with MBR disks, the first sector (LBA 0) of GPT is reserved for MBR data and it is called “protective MBR”.
- The *primary GPT header* begins on the second logical block (LBA 1) of the device. The header contains the disk GUID, the location of the primary partition table, the location of the secondary GPT header, and CRC32 checksums of itself and the primary partition table. It also specifies the number of partition entries of the table.
- The *primary GPT table* includes, by default, 128 partition entries, each with an entry size 128 bytes, its partition type GUID and unique partition GUID.
- The *secondary GPT table* is identical to the primary GPT table. It is used mainly as a backup table for recovery in case the primary partition table is corrupted.
- The *secondary GPT header* is located on the last logical sector of the disk and it can be used to recover GPT information in case the primary header is corrupted. It contains the disk GUID, the location of the secondary partition table and the primary GPT header, CRC32 checksums of itself and the secondary partition table, and the number of possible partition entries.

Important

There must be a BIOS boot partition for the boot loader to be installed successfully onto a disk that contains a GPT (GUID Partition Table). This includes disks initialized by **Anaconda**. If the disk already contains a BIOS boot partition, it can be reused.

B.2. Strategies for Disk Repartitioning

There are several different ways that a disk can be repartitioned. This section discusses the following possible approaches:

- Unpartitioned free space is available
- An unused partition is available
- Free space in an actively used partition is available

Note that this section discusses the aforementioned concepts only theoretically and it does not include any procedures showing how to perform disk repartitioning step-by-step. Such detailed information are beyond the scope of this document.

Note

Keep in mind that the following illustrations are simplified in the interest of clarity and do not reflect the exact partition layout that you encounter when actually installing Fedora.

B.2.1. Using Unpartitioned Free Space

In this situation, the partitions already defined do not span the entire hard disk, leaving unallocated space that is not part of any defined partition. *Figure B.8, "Disk Drive with Unpartitioned Free Space"*, shows what this might look like.

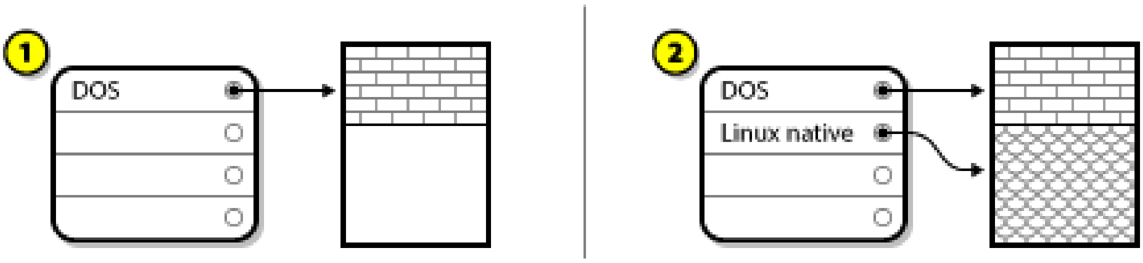


Figure B.8. Disk Drive with Unpartitioned Free Space

In the above example, *1* represents an undefined partition with unallocated space and *2* represents a defined partition with allocated space.

An unused hard disk also falls into this category. The only difference is that *all* the space is not part of any defined partition.

In any case, you can create the necessary partitions from the unused space. Unfortunately, this scenario, although very simple, is not very likely (unless you have just purchased a new disk just for Fedora). Most pre-installed operating systems are configured to take up all available space on a disk drive (see *Section B.2.3, "Using Free Space from an Active Partition"*).

B.2.2. Using Space from an Unused Partition

In this case, maybe you have one or more partitions that you do not use any longer. *Figure B.9, "Disk Drive with an Unused Partition"*, illustrates such a situation.

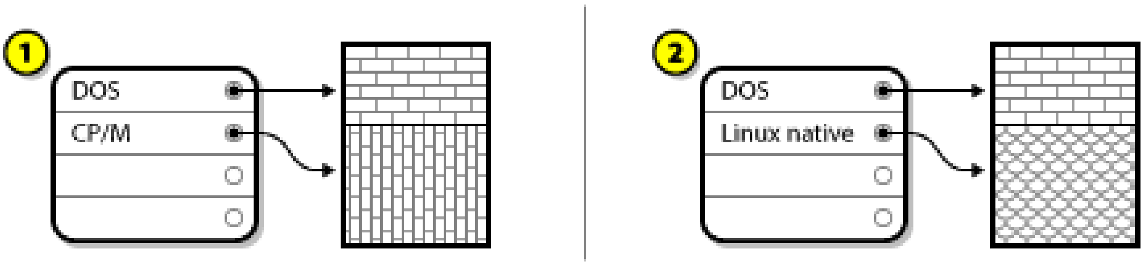


Figure B.9. Disk Drive with an Unused Partition

In the above example, *1* represents an unused partition and *2* represents reallocating an unused partition for Linux.

In this situation, you can use the space allocated to the unused partition. You first must delete the partition and then create the appropriate Linux partition(s) in its place. You can delete the unused partition and manually create new partitions during the installation process.

B.2.3. Using Free Space from an Active Partition

This is the most common situation. It is also, unfortunately, the hardest to handle. The main problem is that, even if you have enough free space, it is presently allocated to a partition that is already in use. If you purchased a computer with pre-installed software, the hard disk most likely has one massive partition holding the operating system and data.

Aside from adding a new hard drive to your system, you have two choices:

Destructive Repartitioning

In this case, the single large partition is deleted and several smaller ones are created instead. Any data held in the original partition is destroyed. This means that making a complete backup is necessary. It is highly recommended to make two backups, use verification (if available in your backup software), and try to read data from the backup *before* deleting the partition.

Warning

If an operating system was installed on that partition, it must be reinstalled if you want to use that system as well. Be aware that some computers sold with pre-installed operating systems may not include the installation media to reinstall the original operating system. You should check whether this applies to your system is *before* you destroy your original partition and its operating system installation.

After creating a smaller partition for your existing operating system, you can reinstall software, restore your data, and start the installation. *Figure B.10, "Disk Drive Being Destructively Repartitioned"* shows this being done.

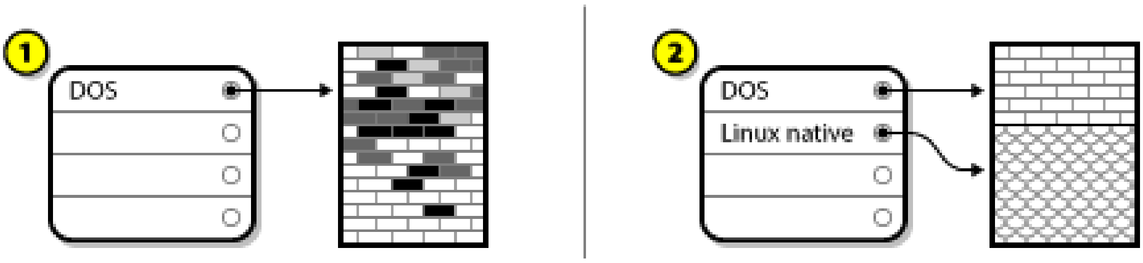


Figure B.10. Disk Drive Being Destructively Repartitioned

In the above example, *1* represents before and *2* represents after.

Warning

Any data previously present in the original partition is lost.

Non-Destructive Repartitioning

With non-destructive repartitioning you execute a program that makes a big partition smaller without losing any of the files stored in that partition. This method is usually reliable, but can be very time-consuming on large drives.

While the process of non-destructive repartitioning is rather straightforward, there are three steps involved:

1. Compress and backup existing data
2. Resize the existing partition
3. Create new partition(s)

Each step is described further in more detail.

B.2.3.1. Compress Existing Data

As the following figure shows, the first step is to compress the data in your existing partition. The reason for doing this is to rearrange the data such that it maximizes the available free space at the "end" of the partition.

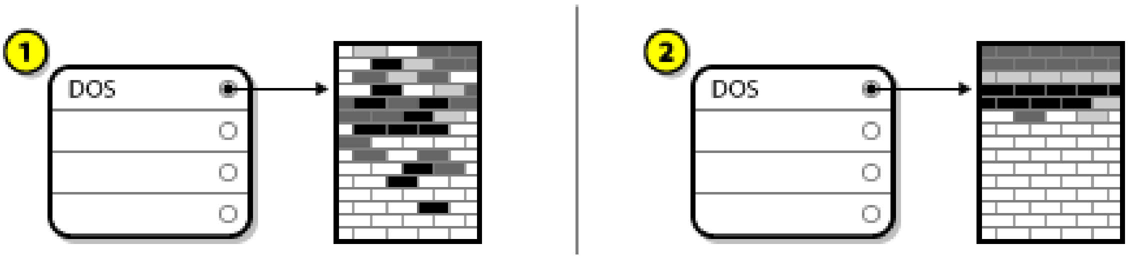


Figure B.11. Disk Drive Being Compressed

In the above example, *1* represents before and *2* represents after.

This step is crucial. Without it, the location of the data could prevent the partition from being resized to the extent desired. Note also that, for one reason or another, some data cannot be moved. If this is the case (and it severely restricts the size of your new partition(s)), you may be forced to destructively repartition your disk.

B.2.3.2. Resize the Existing Partition

Figure B.12, "Disk Drive with Partition Resized" shows the actual resizing process. While the actual result of the resizing operation varies depending on the software used, in most cases the newly freed space is used to create an unformatted partition of the same type as the original partition.

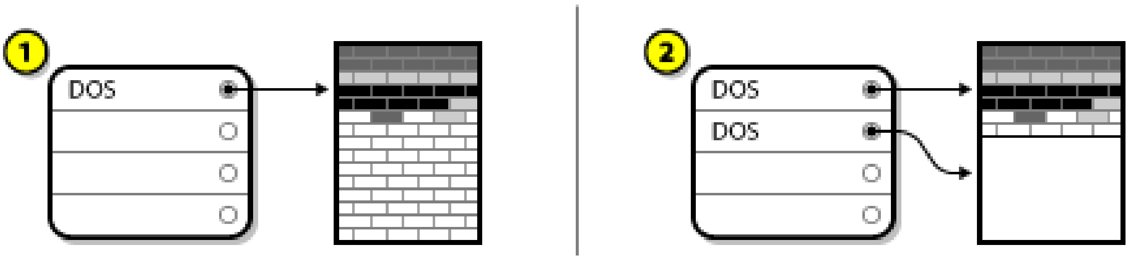


Figure B.12. Disk Drive with Partition Resized

In the above example, *1* represents before and *2* represents after.

It is important to understand what the resizing software you use does with the newly freed space, so that you can take the appropriate steps. In the case illustrated here, it would be best to delete the new DOS partition and create the appropriate Linux partition(s).

B.2.3.3. Create new partition(s)

As the previous step implied, it may or may not be necessary to create new partitions. However, unless your resizing software is Linux-aware, it is likely that you must delete the partition that was created during the resizing process. *Figure B.13, “Disk Drive with Final Partition Configuration”*, shows this being done.

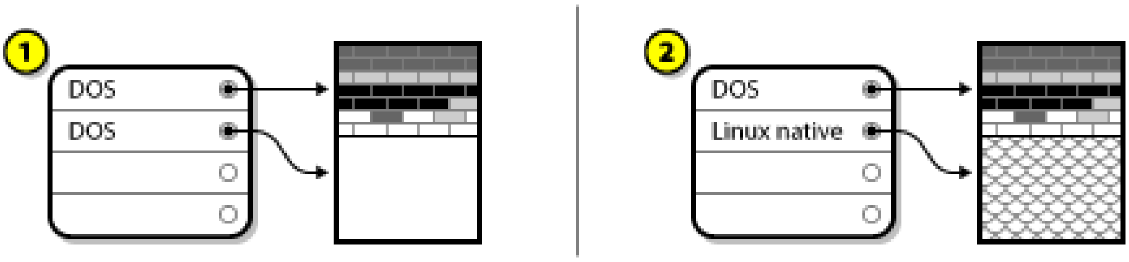


Figure B.13. Disk Drive with Final Partition Configuration

In the above example, *1* represents before and *2* represents after.

B.3. Partition Naming Schemes and Mount Points

A common source of confusion for users unfamiliar with Linux is the matter of how partitions are used and accessed by the Linux operating system. In DOS/Windows, it is relatively simple: Each partition gets a "drive letter." You then use the correct drive letter to refer to files and directories on its corresponding partition. This is entirely different from how Linux deals with partitions and, for that matter, with disk storage in general. This section describes the main principles of partition naming scheme and the way how partitions are accessed in Fedora.

B.3.1. Partition Naming Scheme

Linux uses a naming scheme that is file-based, with file names in the form of **`/dev/`*`xxyN`***.

Device and partition names consist of the following:

`/dev/`

This is the name of the directory in which all device files reside. Because partitions reside on hard disks, and hard disks are devices, the files representing all possible partitions reside in **`/dev/`**.

`xx`

The first two letters of the partition name indicate the type of device on which the partition resides, usually **sd**.

y

This letter indicates which device the partition is on. For example, **/dev/sda** for the first hard disk, **/dev/sdb** for the second, and so on.

N

The final number denotes the partition. The first four (primary or extended) partitions are numbered **1** through **4**. Logical partitions start at **5**. So, for example, **/dev/sda3** is the third primary or extended partition on the first hard disk, and **/dev/sdb6** is the second logical partition on the second hard disk.

Note

Even if your system can identify and refer to all types of disk partitions, it might not be able to *read* the file system and therefore access stored data on every partition type. However, in many cases, it is possible to successfully access data on a partition dedicated to another operating system.

B.3.2. Disk Partitions and Mount Points

Each partition is used to form part of the storage necessary to support a single set of files and directories. This is done by associating a partition with a directory through a process known as *mounting*. Mounting a partition makes its storage available starting at the specified directory (known as a *mount point*).

For example, if partition **/dev/sda5** is mounted on **/usr/**, that would mean that all files and directories under **/usr/** physically reside on **/dev/sda5**. So the file **/usr/share/doc/FAQ/txt/Linux-FAQ** would be stored on **/dev/sda5**, while the file **/etc/gdm/custom.conf** would not.

Continuing the example, it is also possible that one or more directories below **/usr/** would be mount points for other partitions. For instance, a partition (say, **/dev/sda7**) could be mounted on **/usr/local/**, meaning that **/usr/local/man/whatis** would then reside on **/dev/sda7** rather than **/dev/sda5**.

B.3.3. How Many Partitions?

At this point in the process of preparing to install Fedora, you should give some consideration to the number and size of the partitions to be used by your new operating system. However, there is no one right answer to this question. It depends on your needs and requirements.

Unless you have a reason for doing otherwise, you should *at least* create the following partitions: **swap**, **/boot/**, and **/** (root).

See *Section 5.4.10.6, “Recommended Partitioning Scheme”* for more information about the recommended partitioning scheme.

Appendix C. Understanding LVM

LVM (Logical Volume Management) partitions provide a number of advantages over standard partitions. LVM partitions are formatted as *physical volumes*. One or more physical volumes are combined to form a *volume group*. Each volume group's total storage is then divided into one or more *logical volumes*. The logical volumes function much like standard partitions. They have a file system type, such as `ext4`, and a mount point.

Important

The boot loader cannot read LVM volumes. You must make a standard, non-LVM disk partition for your **`/boot`** partition.

By default, the installation process always creates the **`/`** and **`swap`** partitions within LVM volumes, with a separate **`/boot`** on a standard partition. See *Section 5.4.10.6, “Recommended Partitioning Scheme”* for details.

To understand LVM better, imagine the physical volume as a pile of *blocks*. A block is a storage unit used to store data. Several piles of blocks can be combined to make a much larger pile, just as physical volumes are combined to make a volume group. The resulting pile can be subdivided into several smaller piles of arbitrary size, just as a volume group is allocated to several logical volumes.

An administrator may grow or shrink logical volumes without destroying data, unlike standard disk partitions. If the physical volumes in a volume group are on separate drives or RAID arrays then administrators may also spread a logical volume across the storage devices.

You may lose data if you shrink a logical volume to a smaller capacity than the data on the volume requires. To ensure maximum flexibility, create logical volumes to meet your current needs, and leave excess storage capacity unallocated. You may safely grow logical volumes to use unallocated space, as your needs dictate.

Appendix D. Revision History

Note that revision numbers relate to the edition of this manual, not to version numbers of Fedora.

Revision 1.2-0 Mon May 25 2015 **Petr Bokoč** *pbokoc@redhat.cogm*

Fedora 22 Final release
References to **Yum** replaced with **DNF** (the new default package manager)
Updates in boot options: **`inst.dnf`** replaced with **`inst.nodnf`**, added **`inst.kdump_addon=`**
Updates in Kickstart: **`%anaconda`**, **`%addon`**, **`sshkey`**, **`pwpolicy`**
Added the **Kdump** screen to the GUI docs

Revision 1.1-2 Sun Dec 21 2014 **Zach Oglesby** *zach@oglesby.co*

Fixed Windows checksum steps, BZ#1175759

Revision 1.1-1 Mon Dec 15 2014 **Petr Bokoč** *pbokoc@redhat.com*

Fixes in Boot Options: inst.headless is deprecated, inst.askmethod moved from removed to Installation Source
Parts of the network boot setup chapter have been updated with correct URLs
Updates in Recommended Partitioning Scheme and Advice on Partitions in the Manual Partitioning section of the installation chapter

Revision 1.1-0 Mon Dec 8 2014 **Petr Bokoč** *pbokoc@redhat.com*

Publishing for Fedora 21

Revision 1.0-1 Mon Dec 8 2014 **Pete Travis** *immanetize@fedoraproject.org*

added info on media types

Revision 1.0-0 Tue Dec 17 2013 **Petr Bokoč** *pbokoc@redhat.com*

Publishing for Fedora 20

Index