Red Hat Enterprise Linux 9

Boot options for RHEL Installer

Customizing the installation program's behavior by specifying boot options

Red Hat Enterprise Linux 9 Boot options for RHEL Installer

Customizing the installation program's behavior by specifying boot options

Legal Notice

Abstract

RHEL installation system includes a range of boot options for administrators. By using these options, you can modify the default behavior of the installation program by enabling or disabling certain functions to customize the installation program.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message .

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Log in to the Jira website.

2. Click **Create** in the top navigation bar

3. Enter a descriptive title in the **Summary** field.

4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.

5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. BOOT OPTIONS

This section contains information about some of the boot options that you can use to modify the default behavior of the installation program. For a full list of boot options, see the upstream boot option content.

1.1. TYPES OF BOOT OPTIONS

The two types of boot options are those with an equals "=" sign, and those without an equals "=" sign. Boot options are appended to the boot command line and you can append multiple options separated by space. Boot options that are specific to the installation program always start with **inst**.

Options with an equals "=" sign

You must specify a value for boot options that use the **=** symbol. For example, the **inst.vncpassword=** option must contain a value, in this example, a password. The correct syntax for this example is **inst.vncpassword=password**.

Options without an equals "=" sign

This boot option does not accept any values or parameters. For example, the **rd.live.check** option forces the installation program to verify the installation media before starting the installation. If this boot option is present, the installation program performs the verification and if the boot option is not present, the verification is skipped.

1.2. EDITING BOOT OPTIONS

This section describes different ways to edit boot options from the boot menu. The boot menu opens after you boot the installation media.

1.2.1. Editing the boot: prompt in BIOS

When using the **boot:** prompt, the first option must always specify the installation program image file that you want to load. In most cases, you can specify the image using the keyword. You can specify additional options according to your requirements.

Prerequisites

- You have created bootable installation media (USB, CD or DVD).

- You have booted the installation from the media, and the installation boot menu is open.

Procedure

1. With the boot menu open, press the **Esc** key on your keyboard.

2. The **boot:** prompt is now accessible.

3. Press the **Tab** key on your keyboard to display the help commands.

4. Press the **Enter** key on your keyboard to start the installation with your options. To return from the **boot:** prompt to the boot menu, restart the system and boot from the installation media again.

NOTE

The **boot:** prompt also accepts **dracut** kernel options. A list of options is available in the **dracut.cmdline(7)** man page.

1.2.2. Editing predefined boot options using the > prompt

In BIOS-based AMD64 and Intel 64 systems, you can use the **>** prompt to edit predefined boot options. To display a full set of options, select **Test this media and install RHEL 9** from the boot menu.

Prerequisites

- You have created bootable installation media (USB, CD or DVD).

- You have booted the installation from the media, and the installation boot menu is open.

Procedure

1. From the boot menu, select an option and press the **Tab** key on your keyboard. The **>** prompt is accessible and displays the available options.

2. Append the options that you require to the **>** prompt.

3. Press **Enter** to start the installation.

4. Press **Esc** to cancel editing and return to the boot menu.

1.2.3. Editing the GRUB2 menu for the UEFI-based systems

The GRUB2 menu is available on UEFI-based AMD64, Intel 64, and 64-bit ARM systems.

Prerequisites

- You have created bootable installation media (USB, CD or DVD).

- You have booted the installation from the media, and the installation boot menu is open.

Procedure

1. From the boot menu window, select the required option and press **e**.

2. On UEFI systems, the kernel command line starts with **linuxefi**. Move the cursor to the end of the **linuxefi** kernel command line.

3. Edit the parameters as required. For example, to configure one or more network interfaces, add the **ip=** parameter at the end of the **linuxefi** kernel command line, followed by the required value.

4. When you finish editing, press **Ctrl**+**X** to start the installation using the specified options.

1.3. INSTALLATION SOURCE BOOT OPTIONS

This section describes various installation source boot options.

inst.repo=

The **inst.repo=** boot option specifies the installation source, that is, the location providing the package repositories and a valid **.treeinfo** file that describes them. For example: **inst.repo=cdrom**. The target of the **inst.repo=** option must be one of the following installation media:

- an installable tree, which is a directory structure containing the installation program images, packages, and repository data as well as a valid **.treeinfo** file

- a DVD (a physical disk present in the system DVD drive)

- an ISO image of the full Red Hat Enterprise Linux installation DVD, placed on a disk or a network location accessible to the system.
  Use the **inst.repo=** boot option to configure different installation methods using different formats. The following table contains details of the **inst.repo=** boot option syntax:

Table 1.1. Types and format for the inst.repo= boot option and installation source

Source type	Boot option format	Source format
CD/DVD drive	**inst.repo=cdrom:*<device>***	Installation DVD as a physical disk. [a]
Mountable device (HDD and USB stick)	**inst.repo=hd:*<device>*:/*<path>***	Image file of the installation DVD.
NFS Server	**inst.repo=nfs: [*options*:]*<server>*:/*<path>***	Image file of the installation DVD, or an installation tree, which is a complete copy of the directories and files on the installation DVD. [b]
HTTP Server	**inst.repo=http://*<host>*/*<path>***	Installation tree that is a complete copy of the directories and files on the installation DVD.
HTTPS Server	**inst.repo=https://*<host>*/*<path>***	
FTP Server	**inst.repo=ftp://*<username>*:*<password>*@*<host>*/*<path>***	
HMC	**inst.repo=hmc**	

[a] If *device* is left out, installation program automatically searches for a drive containing the installation DVD.

[b] The NFS Server option uses NFS protocol version 3 by default. To use a different version, add **nfsvers=*X*** to *options*, replacing *X* with the version number that you want to use.

Set disk device names with the following formats:

- Kernel device name, for example /**dev**/**sda1** or **sdb2**

- File system label, for example **LABEL=Flash** or **LABEL=RHEL8**

- File system UUID, for example **UUID=8176c7bf-04ff-403a-a832-9557f94e61db**

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

inst.addrepo=

Use the **inst.addrepo=** boot option to add an additional repository that you can use as another installation source along with the main repository (**inst.repo=**). You can use the **inst.addrepo=** boot option multiple times during one boot. The following table contains details of the **inst.addrepo=** boot option syntax.

> **NOTE**
>
> The **REPO_NAME** is the name of the repository and is required in the installation process. These repositories are only used during the installation process; they are not installed on the installed system.

For more information about unified ISO, see Unified ISO.

Table 1.2. Installation sources and boot option format

Installation source	Boot option format	Additional information
Installable tree at a URL	**inst.addrepo=REPO_NAME, [http,https,ftp]://<host>/<path>**	Looks for the installable tree at a given URL.
Installable tree at an NFS path	**inst.addrepo=REPO_NAME,nfs://<server>:/<path>**	Looks for the installable tree at a given NFS path. A colon is required after the host. The installation program passes everything after **nfs://** directly to the mount command instead of parsing URLs according to RFC 2224.

Installation source	Boot option format	Additional information
Installable tree in the installation environment	**inst.addrepo=REPO_NAME,file://***<path>*	Looks for the installable tree at the given location in the installation environment. To use this option, the repository must be mounted before the installation program attempts to load the available software groups. The benefit of this option is that you can have multiple repositories on one bootable ISO, and you can install both the main repository and additional repositories from the ISO. The path to the additional repositories is **/run/install/source/REPO_ISO_PATH**. Additionally, you can mount the repository directory in the **%pre** section in the Kickstart file. The path must be absolute and start with /, for example **inst.addrepo=REPO_NAME,file:///***<path>*
Disk	**inst.addrepo=REPO_NAME,hd:***<device>***:***<path>*	Mounts the given *<device>* partition and installs from the ISO that is specified by the *<path>*. If the *<path>* is not specified, the installation program looks for a valid installation ISO on the *<device>*. This installation method requires an ISO with a valid installable tree.

inst.stage2=

The **inst.stage2=** boot option specifies the location of the installation program's runtime image. This option expects the path to a directory that contains a valid **.treeinfo** file and reads the runtime image location from the **.treeinfo** file. If the **.treeinfo** file is not available, the installation program attempts to load the image from **images/install.img**.
When you do not specify the **inst.stage2** option, the installation program attempts to use the location specified with the **inst.repo** option.

Use this option when you want to manually specify the installation source in the installation program at a later time. For example, when you want to select the Content Delivery Network (CDN) as an installation source. The installation DVD and Boot ISO already contain a suitable **inst.stage2** option to boot the installation program from the respective ISO.

If you want to specify an installation source, use the **inst.repo=** option instead.

> **NOTE**
>
> By default, the **inst.stage2=** boot option is used on the installation media and is set to a specific label; for example, **inst.stage2=hd:LABEL=RHEL-x-0-0-BaseOS-x86_64**. If you modify the default label of the file system that contains the runtime image, or if you use a customized procedure to boot the installation system, verify that the **inst.stage2=** boot option is set to the correct value.

inst.noverifyssl

Use the **inst.noverifyssl** boot option to prevent the installer from verifying SSL certificates for all HTTPS connections with the exception of additional Kickstart repositories, where **--noverifyssl** can be set per repository.

For example, if your remote installation source is using self-signed SSL certificates, the **inst.noverifyssl** boot option enables the installer to complete the installation without verifying the SSL certificates.

Example when specifying the source using inst.stage2=

```
inst.stage2=https://hostname/path_to_install_image/ inst.noverifyssl
```

Example when specifying the source using inst.repo=

```
inst.repo=https://hostname/path_to_install_repository/ inst.noverifyssl
```

inst.stage2.all

Use the **inst.stage2.all** boot option to specify several HTTP, HTTPS, or FTP sources. You can use the **inst.stage2=** boot option multiple times with the **inst.stage2.all** option to fetch the image from the sources sequentially until one succeeds. For example:

```
inst.stage2.all
inst.stage2=http://hostname1/path_to_install_tree/
inst.stage2=http://hostname2/path_to_install_tree/
inst.stage2=http://hostname3/path_to_install_tree/
```

inst.dd=

The **inst.dd=** boot option is used to perform a driver update during the installation. For more information about how to update drivers during installation, see the *Performing an advanced RHEL 9 installation* document.

inst.repo=hmc

This option eliminates the requirement of an external network setup and expands the installation options. When booting from a Binary DVD, the installation program prompts you to enter additional kernel parameters. To set the DVD as an installation source, append the **inst.repo=hmc** option to the kernel parameters. The installation program then enables support element (SE) and hardware management console (HMC) file access, fetches the images for stage2 from the DVD, and provides access to the packages on the DVD for software selection.

inst.proxy=

The **inst.proxy=** boot option is used when performing an installation from a HTTP, HTTPS, and FTP protocol. For example:

```
[PROTOCOL://][USERNAME[:PASSWORD]@]HOST[:PORT]
```

inst.nosave=

Use the **inst.nosave=** boot option to control the installation logs and related files that are not saved to the installed system, for example **input_ks**, **output_ks**, **all_ks**, **logs** and **all**. You can combine multiple values separated by a comma. For example,

> inst.nosave=Input_ks,logs

> **NOTE**
>
> The **inst.nosave** boot option is used for excluding files from the installed system that cannot be removed by a Kickstart %post script, such as logs and input/output Kickstart results.

input_ks

Disables the ability to save the input Kickstart results.

output_ks

Disables the ability to save the output Kickstart results generated by the installation program.

all_ks

Disables the ability to save the input and output Kickstart results.

logs

Disables the ability to save all installation logs.

all

Disables the ability to save all Kickstart results, and all logs.

inst.multilib

Use the **inst.multilib** boot option to set DNF's **multilib_policy** to **all**, instead of **best**.

inst.memcheck

The **inst.memcheck** boot option performs a check to verify that the system has enough RAM to complete the installation. If there is not enough RAM, the installation process is stopped. The system check is approximate and memory usage during installation depends on the package selection, user interface, for example graphical or text, and other parameters.

inst.nomemcheck

The **inst.nomemcheck** boot option does not perform a check to verify if the system has enough RAM to complete the installation. Any attempt to perform the installation with less than the recommended minimum amount of memory is unsupported, and might result in the installation process failing.

1.4. NETWORK BOOT OPTIONS

If your scenario requires booting from an image over the network instead of booting from a local image, you can use the following options to customize network booting.

> **NOTE**
>
> Initialize the network with the **dracut** tool. For complete list of **dracut** options, see the **dracut.cmdline(7)** man page.

ip=

Use the **ip=** boot option to configure one or more network interfaces. To configure multiple interfaces, use one of the following methods;

- use the **ip** option multiple times, once for each interface; to do so, use the **rd.neednet=1** option, and specify a primary boot interface using the **bootdev** option.

- use the **ip** option once, and then use Kickstart to set up further interfaces. This option accepts several different formats. The following tables contain information about the most common options.

In the following tables:

- The **ip** parameter specifies the client IP address and **IPv6** requires square brackets, for example 192.0.2.1 or [2001:db8::99].

- The **gateway** parameter is the default gateway. **IPv6** requires square brackets.

- The **netmask** parameter is the netmask to be used. This can be either 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.

Table 1.3. Boot option formats to configure the network interface

Boot option format	Configuration method
ip=*method*	Automatic configuration of any interface
ip=*interface:method*	Automatic configuration of a specific interface
ip=*ip::gateway:netmask:hostname:interface*:none	Static configuration, for example, IPv4: **ip=192.0.2.1::192.0.2.254:255.255.255.0:server.example.com:enp1s0:none** IPv6: **ip=[2001:db8::1]::[2001:db8::fffe]:64:server.example.com:enp1s0:none**
ip=*ip::gateway:netmask:hostname:interface*:method:mtu	Automatic configuration of a specific interface with an override

Configuration methods for the automatic interface

The method **automatic configuration of a specific interface with an override** opens the interface using the specified method of automatic configuration, such as **dhcp**, but overrides the automatically obtained IP address, gateway, netmask, host name or other specified parameters. All parameters are optional, so specify only the parameters that you want to override.

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

DHCP

   dhcp

IPv6 DHCP

dhcp6

IPv6 automatic configuration

auto6

iSCSI Boot Firmware Table (iBFT)

ibft

> **NOTE**
>
> - If you use a boot option that requires network access, such as **inst.ks=http://host/path**, without specifying the **ip** option, the default value of the **ip** option is **ip=dhcp**..
>
> - To connect to an iSCSI target automatically, activate a network device for accessing the target by using the **ip=ibft** boot option.

nameserver=

The **nameserver=** option specifies the address of the name server. You can use this option multiple times.

> **NOTE**
>
> The **ip=** parameter requires square brackets. However, an IPv6 address does not work with square brackets. An example of the correct syntax to use for an IPv6 address is **nameserver=2001:db8::1**.

bootdev=

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

ifname=

The **ifname=** options assigns an interface name to a network device with a given MAC address. You can use this option multiple times. The syntax is **ifname=interface:MAC**. For example:

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

> **NOTE**
>
> The **ifname=** option is the only supported way to set custom network interface names during installation.

inst.dhcpclass=

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

inst.waitfornet=

Using the **inst.waitfornet=SECONDS** boot option causes the installation system to wait for network connectivity before installation. The value given in the **SECONDS** argument specifies the maximum amount of time to wait for network connectivity before timing out and continuing the installation process even if network connectivity is not present.

vlan=

Use the **vlan=** option to configure a Virtual LAN (VLAN) device on a specified interface with a given name. The syntax is **vlan=name:interface**. For example:

> vlan=vlan5:enp0s1

This configures a VLAN device named **vlan5** on the **enp0s1** interface. The name can take the following forms:

- VLAN_PLUS_VID: **vlan0005**

- VLAN_PLUS_VID_NO_PAD: **vlan5**

- DEV_PLUS_VID: **enp0s1.0005**

- DEV_PLUS_VID_NO_PAD: **enp0s1.5**

  bond=

  Use the **bond=** option to configure a bonding device with the following syntax: **bond=name[:interfaces][:options]**. Replace *name* with the bonding device name, *interfaces* with a comma-separated list of physical (Ethernet) interfaces, and *options* with a comma-separated list of bonding options. For example:

  > bond=bond0:enp0s1,enp0s2:mode=active-backup,tx_queues=32,downdelay=5000

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

team=

Use the **team=** option to configure a team device with the following syntax: **team=name:interfaces**. Replace *name* with the desired name of the team device and *interfaces* with a comma-separated list of physical (Ethernet) devices to be used as underlying interfaces in the team device. For example:

> team=team0:enp0s1,enp0s2

> **IMPORTANT**
>
> Network teaming is deprecated in Red Hat Enterprise Linux 9. Consider using the network bonding driver as an alternative. For details, see Configuring network bonding.

bridge=

Use the **bridge=** option to configure a bridge device with the following syntax: **bridge=name:interfaces**. Replace *name* with the desired name of the bridge device and *interfaces* with a comma-separated list of physical (Ethernet) devices to be used as underlying interfaces in the bridge device. For example:

> bridge=bridge0:enp0s1,enp0s2

Additional resources

- *Configuring and managing networking*

1.5. CONSOLE BOOT OPTIONS

This section describes how to configure boot options for your console, monitor display, and keyboard.

console=

Use the **console=** option to specify a device that you want to use as the primary console. For example, to use a console on the first serial port, use **console=ttyS0**. When using the **console=** argument, the installation starts with a text UI. If you must use the **console=** option multiple times, the boot message is displayed on all specified console. However, the installation program uses only the last specified console. For example, if you specify **console=ttyS0 console=ttyS1**, the installation program uses **ttyS1**.

inst.lang=

Use the **inst.lang=** option to set the language that you want to use during the installation. To view the list of locales, enter the command **locale -a | grep _** or the **localectl list-locales | grep _** command.

inst.geoloc=

Use the **inst.geoloc=** option to configure geolocation usage in the installation program. Geolocation is used to preset the language and time zone, and uses the following syntax: **inst.geoloc=value**. The **value** can be any of the following parameters:

- Disable geolocation: **inst.geoloc=0**

- Use the Fedora GeoIP API: **inst.geoloc=provider_fedora_geoip**. This option is deprecated.

- Use the Hostip.info GeoIP API: **inst.geoloc=provider_hostip**. This option is deprecated.

inst.keymap=

Use the **inst.keymap=** option to specify the keyboard layout to use for the installation.

inst.cmdline

Use the **inst.cmdline** option to force the installation program to run in command-line mode. This mode does not allow any interaction, and you must specify all options in a Kickstart file or on the command line.

inst.graphical

Use the **inst.graphical** option to force the installation program to run in graphical mode. The graphical mode is the default.

inst.text

Use the **inst.text** option to force the installation program to run in text mode instead of graphical mode.

inst.noninteractive

Use the **inst.noninteractive** boot option to run the installation program in a non-interactive mode. User interaction is not permitted in the non-interactive mode, and **inst.noninteractive** you can use the **inst.nointeractive** option with a graphical or text installation. When you use the **inst.noninteractive** option in text mode, it behaves the same as the **inst.cmdline** option.

inst.resolution=

Use the **inst.resolution=** option to specify the screen resolution in graphical mode. The format is **NxM**, where *N* is the screen width and *M* is the screen height (in pixels). The recommended resolution is 1024x768.

inst.vnc

Use the **inst.vnc** option to run the graphical installation using Virtual Network Computing (VNC). You must use a VNC client application to interact with the installation program. When VNC sharing is enabled, multiple clients can connect. A system installed using VNC starts in text mode.

inst.vncpassword=

Use the **inst.vncpassword=** option to set a password on the VNC server that is used by the installation program.

inst.vncconnect=

Use the **inst.vncconnect=** option to connect to a listening VNC client at the given host location, for example, **inst.vncconnect=<host>[:<port>]** The default port is 5900. You can use this option by entering the command **vncviewer -listen**.

inst.xdriver=

Use the **inst.xdriver=** option to specify the name of the X driver to use both during installation and on the installed system.

inst.usefbx

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

modprobe.blacklist=

Use the **modprobe.blacklist=** option to blocklist or completely disable one or more drivers. Drivers (mods) that you disable using this option cannot load when the installation starts. After the installation finishes, the installed system retains these settings. You can find a list of the blocklisted drivers in the **/etc/modprobe.d/** directory. Use a comma-separated list to disable multiple drivers. For example:

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

inst.xtimeout=

Use the **inst.xtimeout=** option to specify the timeout in seconds for starting X server.

inst.sshd

Use the **inst.sshd** option to start the **sshd** service during installation, so that you can connect to the system during the installation using SSH, and monitor the installation progress. For more information about SSH, see the **ssh(1)** man page. By default, the **sshd** option is automatically started only on the 64-bit IBM Z architecture. On other architectures, **sshd** is not started unless you use the **inst.sshd** option.
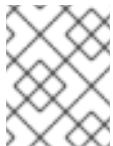
> **NOTE**
>
> During installation, the root account has no password by default. You can set a root password during installation with the **sshpw** Kickstart command.

inst.kdump_addon=

Use the **inst.kdump_addon=** option to enable or disable the Kdump configuration screen (add-on) in the installation program. This screen is enabled by default; use **inst.kdump_addon=off** to disable it. Disabling the add-on disables the Kdump screens in both the graphical and text-based interface as well as the **%addon com_redhat_kdump** Kickstart command.

1.6. DEBUG BOOT OPTIONS

This section describes the options you can use when debugging issues.

inst.rescue

Use the **inst.rescue** option to run the rescue environment for diagnosing and fixing systems. For example, you can repair a filesystem in rescue mode .

inst.updates=

Use the **inst.updates=** option to specify the location of the **updates.img** file that you want to apply during installation. The **updates.img** file can be derived from one of several sources.

Table 1.4. **updates.img** file sources

Source	Description	Example
Updates from a network	Specify the network location of **updates.img**. This does not require any modification to the installation tree. To use this method, edit the kernel command line to include **inst.updates**.	**inst.updates=http://website. com/path/to/updates.img**.
Updates from a disk image	Save an **updates.img** on a floppy drive or a USB key. This can be done only with an **ext2** filesystem type of **updates.img**. To save the contents of the image on your floppy drive, insert the floppy disc and run the command.	**dd if=updates.img of=/dev/fd0 bs=72k count=20**. To use a USB key or flash media, replace **/dev/fd0** with the device name of your USB flash drive.
Updates from an installation tree	If you are using a CD, disk, HTTP, or FTP install, save the **updates.img** in the installation tree so that all installations can detect the **.img** file. The file name must be **updates.img**.	For NFS installs, save the file in the **images/** directory, or in the **RHupdates/** directory.

inst.syslog=

Sends log messages to the **syslog** process on the specified host when the installation starts. You can use **inst.syslog=** only if the remote **syslog** process is configured to accept incoming connections.

inst.virtiolog=

Use the **inst.virtiolog=** option to specify which virtio port (a character device at **/dev/virtio-ports/name**) to use for forwarding logs. The default value is **org.fedoraproject.anaconda.log.0**.

rd.live.ram

Copies the **stage 2** image in **images/install.img** into RAM. Note that this increases the memory required for installation by the size of the image which is usually between 400 and 800MB.

inst.nokill

Prevent the installation program from rebooting when a fatal error occurs, or at the end of the installation process. Use it capture installation logs which would be lost upon reboot.

inst.noshell

Prevent a shell on terminal session 2 (tty2) during installation.

inst.notmux

Prevent the use of tmux during installation. The output is generated without terminal control characters and is meant for non-interactive uses.

inst.remotelog=

Sends all the logs to a remote **host:port** using a TCP connection. The connection is retired if there is no listener and the installation proceeds as normal.

1.7. STORAGE BOOT OPTIONS

This section describes the options you can specify to customize booting from a storage device.

inst.nodmraid

Disables **dmraid** support.

> **WARNING**
>
> Use this option with caution. If you have a disk that is incorrectly identified as part of a firmware RAID array, it might have some stale RAID metadata on it that must be removed using the appropriate tool such as, **dmraid** or **wipefs**.

inst.nompath

Disables support for multipath devices. Use this option only if your system has a false-positive that incorrectly identifies a normal block device as a multipath device.

> **WARNING**
>
> Use this option with caution. Do not use this option with multipath hardware. Using this option to install to a single path of a multipath device is not supported.

inst.gpt

Forces the installation program to install partition information to a GUID Partition Table (GPT) instead of a Master Boot Record (MBR). This option is not valid on UEFI-based systems, unless they are in BIOS compatibility mode. Normally, BIOS-based systems and UEFI-based systems in BIOS compatibility mode attempt to use the MBR schema for storing partitioning information, unless the disk is 2^32 sectors in size or larger. Disk sectors are typically 512 bytes in size, meaning that this is usually equivalent to 2 TiB. The **inst.gpt** boot option allows a GPT to be written to smaller disks.

inst.wait_for_disks=

Use the **inst.wait_for_disks=** option to specify the number of seconds installation program to wait for disk devices to appear at the beginning of the installation. Use this option when you use the **OEMDRV-labeled** device to automatically load the Kickstart file or the kernel drivers but the device takes longer time to appear during the boot process. By default, installation program waits for **5** seconds. Use **0** seconds to minimize the delay.

1.8. KICKSTART BOOT OPTIONS

This section describes the boot options you can add in the Kickstart file to automate an installation.

inst.ks=

Defines the location of a Kickstart file to use to automate the installation. You can specify locations using any of the **inst.repo** formats. If you specify a device and not a path, the installation program looks for the Kickstart file in /**ks.cfg** on the specified device.

If you use this option without specifying a device, the installation program uses the following value for the option:

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

In the previous 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. The following table contains an example.

Table 1.5. 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

If a volume with a label of **OEMDRV** is present, the installation program attempts to load a Kickstart file named **ks.cfg**. If your Kickstart file is in this location, you do not need to use the **inst.ks=** boot option.

inst.ks.all

Specify the **inst.ks.all** option to sequentially try multiple Kickstart file locations provided by multiple **inst.ks** options. The first successful location is used. This applies only to locations of type **http**, **https** or **ftp**, other locations are ignored.

inst.ks.sendmac

Use the **inst.ks.sendmac** option to add headers to outgoing HTTP requests that contain 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

Use the **inst.ks.sendsn** option to add a header to outgoing HTTP requests. This header contains the system serial number, read from /**sys**/**class**/**dmi**/**id**/**product_serial**. The header has the following syntax:

> X-System-Serial-Number: R8VA23D

Additional resources

- Full list of boot options

1.9. ADVANCED INSTALLATION BOOT OPTIONS

This section contains information about advanced installation boot options.

inst.kexec

Runs the **kexec** system call at the end of the installation, instead of performing a reboot. The **inst.kexec** option loads the new system immediately, and bypasses the hardware initialization normally performed by the BIOS or firmware.
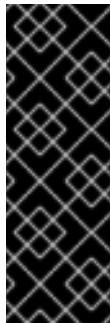
> **IMPORTANT**
>
> This option is deprecated and available as a Technology Preview only. For information about Red Hat scope of support for Technology Preview features, see the Technology Preview Features Support Scope document.
>
> When **kexec** is used, device registers, which would normally be cleared during a full system reboot, might stay filled with data. This can potentially create issues for certain device drivers.

inst.multilib

Configures the system for multilib packages to allow installing 32-bit packages on a 64-bit AMD64 or Intel 64 system. 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, are installed. When you use the **inst.multilib** boot option, packages for 32-bit AMD or Intel systems, marked as i686, are automatically installed.
This applies only to packages directly specified in the **%packages** section. If a package is installed as a dependency, only the exact specified dependency is installed. For example, if you are installing the **bash** package that depends on the **glibc** package, the **bash** package is installed in multiple variants, while the **glibc** package is installed only in variants that the bash package requires.

selinux=0

Disables the use of SELinux in the installation program and the installed system. By default, SELinux operates in permissive mode in the installation program, and in enforcing mode in the installed system.
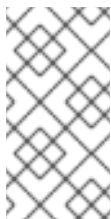
> **NOTE**
>
> The inst.selinux=0 and selinux=0 options are not the same: * inst.selinux=0: disable SELinux only in the installation program. * selinux=0: disable the use of SELinux in the installation program and the installed system. Disabling SELinux causes events not to be logged.

inst.nonibftiscsiboot

Places the boot loader on iSCSI devices that were not configured in the iSCSI Boot Firmware Table (iBFT).

1.10. REMOVED BOOT OPTIONS

This section contains the boot options that have been removed from Red Hat Enterprise Linux.

inst.zram

The **zram.service** cannot be run anymore. See **zram-generator** for more information.

inst.singlelang

The single language mode is not supported anymore.

inst.loglevel

The log level is always set to debug.