Red Hat Enterprise Linux 7

Migration Planning Guide

Key differences between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7

Last Updated: 2020-06-07

Red Hat Enterprise Linux 7 Migration Planning Guide

Key differences between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7

Legal Notice

Abstract

This document provides an overview of changes in Red Hat Enterprise Linux 7 since Red Hat Enterprise Linux 6 to help you evaluate migration to Red Hat Enterprise Linux 7.

Table of Contents

CHAPTER 1. HOW TO UPGRADE

As of Red Hat Enterprise Linux 7.0 GA, there is one supported upgrade path: from the latest version of Red Hat Enterprise Linux 6 to the latest version of Red Hat Enterprise Linux 7.

1.1. HOW TO UPGRADE FROM RED HAT ENTERPRISE LINUX 6

Follow the procedures in this chapter to upgrade to Red Hat Enterprise Linux 7 from Red Hat Enterprise Linux 6. The upgrade process consists of the following steps:

1. Check that Red Hat supports the upgrade of your system. See Section 1.1.1, “Check your support status” for details.

2. Prepare your system for upgrade. See Section 1.1.2, “Prepare your system for upgrade” for details.

3. Check your system for problems that might affect your upgrade. See Section 1.1.3, “Check system upgrade suitability” for details.

4. Upgrade by running the Red Hat Upgrade Tool. See Section 1.1.4, “Upgrade your system” for details.

1.1.1. Check your support status

Upgrading from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7 is supported only if your system meets the following criteria.

- Your system is on the latest version of the Server variant of Red Hat Enterprise Linux 6 for Intel 64 and AMD64 architecture, with all packages up to date. To check, enter the following commands:

  ```
  # cat /etc/redhat-release
  Red Hat Enterprise Linux Server release 6.9 (Santiago)
  # arch
  x86_64
  # yum upgrade -y
  ```

- Your system is registered to receive updates from Subscription Management and not RHN Classic.

- Your system includes only the following package groups:
  - Minimal
  - Base
  - Web Server
  - DHCP Server
  - NFS File Server (**@nfs-server**)
  - Print Server
  - CIFS file server

Remove other package groups before upgrading and reinstall them when your upgrade is complete.

Check the following Knowledgebase Solution for details: https://access.redhat.com/solutions/799813.

1.1.2. Prepare your system for upgrade

Red Hat Enterprise Linux 7 is the first major release to allow in-place upgrades from the previous major version. Upgrades of this magnitude are complex and prone to error. To ensure your upgrade goes as smoothly as possible, some preparations are necessary.

Back up all data

Firstly, back up the entire system to avoid potential data loss, and test that your backup works.

Test first

Before you upgrade a production system, you should clone the system and test the upgrade procedure on the clone. This will allow you to prepare for upgrade without risking the production system.

Convert to Red Hat Subscription Management

Red Hat Enterprise Linux 7 must be registered with the Subscription Management tool (**subscription-manager**) rather than RHN Classic tools like **rhn_register**. See https://access.redhat.com/articles/433903 for details on getting started with Subscription Management.
Before it begins upgrading packages, the **yum upgrade** command outputs a statement about how this system receives updates. Ensure that **subscription-manager** and not RHN is mentioned here.

```
# yum upgrade
Loaded plug-ins: product-id, security, subscription-manager
...
```

If your Red Hat Enterprise Linux 6 system is currently registered to RHN Classic, you must first unregister from RHN Classic by following these instructions: https://access.redhat.com/solutions/11272.

Ensure only supported package groups are installed

This upgrade process supports only the following package groups:

- Minimal
- Base
- Web Server
- DHCP Server
- NFS File Server (**@nfs-server**)
- Print Server
- CIFS file server
  Check which package groups are installed using the **yum grouplist** command. Remove other package groups before upgrading and reinstall them when your upgrade is complete.

Update all packages

Once your system is registered with Subscription Management, ensure that all packages on your system are up to date by running the following commands.

```
# yum update -y
# reboot
```

1.1.3. Check system upgrade suitability

The Preupgrade Assistant checks your Red Hat Enterprise Linux 6 system for anything that might adversely affect the success of your upgrade.

- Install and run the Preupgrade Assistant, **preupg**. See Section 1.1.3.1, “Installing the Preupgrade Assistant” and Section 1.1.3.2, “Running the Preupgrade Assistant” for details.
- Correct any problems identified by the Preupgrade Assistant. See Section 1.1.3.3, “Viewing results and correcting errors” for details.
- Consult the Release Notes, Technical Notes, and Migration Planning Guide to ensure that you are aware of any changes that the Preupgrade Assistant does not assess. The latest versions of these documents are available from https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/.

1.1.3.1. Installing the Preupgrade Assistant

1. Enable the Extras repository
   As root, enter the following command to subscribe your system to the repository containing the Preupgrade Assistant.

   If your system receives updates from Red Hat Subscription Management:

   ```
   # subscription-manager repos --enable rhel-6-server-extras-rpms
   ```

2. Install the tool
   As root, enter the following command to install all the Preupgrade Assistant packages.

   ```
   # yum -y install preupgrade-assistant preupgrade-assistant-el6toel7
   ```

1.1.3.2. Running the Preupgrade Assistant

To run the Preupgrade Assistant, execute the following command as root.

```
# preupg -v
```

This takes a few minutes to complete.

Alternatively, if you have already configured a Preupgrade Assistant Web UI that this system can access, execute the following, replacing *hostname* and *port* with the appropriate values for your Preupgrade Assistant Web UI:

```
# preupg -v -u http://hostname:_port_/submit
```

See Section 1.1.3.2.1, "Configuring the Preupgrade Assistant Web UI" for instructions on setting up the web UI.

1.1.3.2.1. Configuring the Preupgrade Assistant Web UI

Preupgrade Assistant Web UI lets you upload and view pre-upgrade results through a browser-based interface. This is useful when the system you intend to upgrade is headless or does not have a graphical user environment.

> **WARNING**
>
> Preupgrade Assistant Web UI requires a running instance of Apache Web Server (**httpd**) and a number of changes to the **/etc/httpd/conf.d** directory. If you are concerned about exposing data about your system to the network, or you want to avoid adding packages to the system you intend to upgrade, do not follow this procedure. Instead, copy the **/root/preupgrade/result.html** file to a machine with a graphical user interface and view it in a web browser.

1. Install required packages

   ```
   # yum -y install httpd preupgrade-assistant-ui
   ```

   a. Change upload configuration Change from using the default private pre-upgrade configuration to the public configuration.

      ```
      # cd /etc/httpd/conf.d
      # cp 99-preup-httpd.conf.public 99-preup-httpd.conf
      ```

      This makes the Preupgrade Assistant Web UI available to all network interfaces on the local system, through TCP port 8099 by default.

      You can also edit the **NameVirtualHost** variable in the new **/etc/httpd/conf.d/99-preup-httpd.conf** to set a host name, if you want to access Preupgrade Assistant Web UI through a host name like **preupg-ui.example.com:8099** instead of an IP address like **192.168.99.1:8099**.

   b. Change firewall and SELinux configuration Temporarily put SELinux in permissive mode, and allow traffic through TCP port 8099.

      ```
      # setenforce 0
      # iptables -I INPUT -m state --state NEW -p tcp --dport 8099 -j ACCEPT
      ```

   c. Restart the web server

      ```
      # service httpd restart
      ```

2. Configure or disable authentication In a web browser, navigate to **http://192.168.99.1:8099/** (or the host name, if you specified one in Step 2). You are prompted to either enter details to create a new administrative user, or disable authentication.

You can now upload pre-upgrade test results from the system you intend to upgrade by running the **preupg** command with the **-u http://*hostname*:*port*/submit** option on that system.

1.1.3.3. Viewing results and correcting errors

When you run **preupg**, a summary of results is printed to standard output. Detailed results are saved to the **/root/preupgrade** directory as **result.html** by default. You can also upload results to the Preupgrade Assistant Web UI to compare multiple **preupg** runs as you correct potential migration issues.

Regardless of how you view the assessment results, you need to check each item for corrections that need to be made before you upgrade. Each item is printed with an exit code that indicates that item's suitability for in-place upgrade, as described in Table 1.1, "Exit Codes".

Table 1.1. Exit Codes

Exit Code	Definition
PASS	Everything looks fine, and this item is ready to upgrade.
FAIL	Extreme upgrade risk. upgrade is not possible.
NEEDS_ACTION	High upgrade risk. Some administrator action is required before you upgrade.
NEEDS_INSPECTION	Moderate and lower risk. Upgrading should succeed, but may result in a system that is not fully functional.
FIXED	A change required for upgrade was made automatically.
INFORMATIONAL	Useful but noncritical information about this item is available in the report.
NOT_APPLICABLE	Preupgrade Assistant checked for an item that was not installed on your system.
ERROR	Something has probably gone wrong with the pre-upgrade tools. Report this type of problem to Red Hat Support.

Take note of any post upgrade tasks identified by Preupgrade Assistant; you will need to perform these after you run the Red Hat Upgrade Tool.

You should also check the Release Notes, Technical Notes, and Migration Planning Guide for items that the Preupgrade Assistant is unable to detect.

1.1.4. Upgrade your system

After you have corrected all issues reported by the Preupgrade Assistant, you are ready to use the Red Hat Upgrade Tool to upgrade your system.

IMPORTANT

Running the Red Hat Upgrade tool requires you to run the Preupgrade Assistant tool as a prerequisite. If you try to run Red Hat Upgrade on your system without first running the Preupgrade Assistant, it will exit with the following error:

```
preupgrade-assistant has not been run
```

WARNING

Test this process on a non-production system before you perform it on any production system.

1. Install the tool

   ```
   # yum -y install redhat-upgrade-tool
   ```

2. Disable active repositories

   ```
   # yum -y install yum-utils
   # yum-config-manager --disable \*
   ```

3. Perform the upgrade
   The upgrade process requires access to Red Hat Enterprise Linux 7 packages. You can specify the location of a repository on the network or on a mounted device, or an ISO image, as shown below.

   ```
   # redhat-upgrade-tool --network <latest_RHEL_7> --instrepo repo_location
   ```

   ```
   # redhat-upgrade-tool --device device_path
   ```

   ```
   # redhat-upgrade-tool --iso iso_path
   ```

   Some packages that were in the Base package group in Red Hat Enterprise Linux 6 are no longer part of that group in Red Hat Enterprise Linux 7. You may need to configure additional repositories in order to upgrade these packages correctly.

   Refer to https://access.redhat.com/site/solutions/912213 to enable the Extras repository on the yum repository system. Then see https://access.redhat.com/site/solutions/9892 to set up a repository that you can use during your upgrade. The upgrade command for this use case would look similar to the following.

   ```
   # redhat-upgrade-tool --addrepo optional=http://host name/path/to/repo
   ```

   Some packages are not reinstalled during the upgrade process because they have no functionally equivalent replacements in Red Hat Enterprise Linux 7. Red Hat does not provide any support for these packages. To remove these packages at the end of the upgrade process, enter the following command:

```
# redhat-upgrade-tool --cleanup-post
```

4. Reboot
   When prompted, reboot the system.

5. Wait for upgrade to complete
   After your system reboots, upgrade can take several minutes or several hours, depending on the number of packages to install.

6. Perform post upgrade tasks
   Manually perform any post upgrade tasks described in the Preupgrade Assistant assessment result. See Section 1.1.3.3, “Viewing results and correcting errors” for details.

   IMPORTANT

   If Samba is installed on the upgraded host, manually run the **testparm** utility to verify the **/etc/samba/smb.conf** file. If **testparm** reports any configuration errors, you must fix them before you can start Samba.

7. Check system status
   Check that your system’s subscription details have been updated as part of the upgrade process.

```
# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.4
# yum repolist
Loaded plug-ins: product-id, subscription-manager
repo id     repo name                        status
rhel-7-rpms  Red Hat Enterprise Linux 7 Server (RPMs)   4,323
```

   If the list of your repositories did not update correctly, perform the following commands:

```
# subscription-manager remove --all
# subscription-manager unregister
# subscription-manager register
# subscription-manager attach --pool=poolID
# subscription-manager repos --enable=repoID
```

8. Update all packages
   Ensure that all packages are up to date by running the following:

```
# yum upgrade -y
# reboot
```

CHAPTER 2. MAJOR CHANGES AND MIGRATION CONSIDERATIONS

This chapter discusses major changes and features that may affect migration from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7. Read each section carefully for a clear understanding of how your system will be impacted by upgrading to Red Hat Enterprise Linux 7.

2.1. SYSTEM LIMITATIONS

Red Hat Enterprise Linux supported system limitations have changed between version 6 and version 7.

Red Hat Enterprise Linux 7 now requires at least 1 GB of disk space to install. However, Red Hat recommends a minimum of 5 GB of disk space for all supported architectures.

AMD64 and **Intel** 64 systems now require at least 1 GB of memory to run. Red Hat recommends at least 1 GB memory per logical CPU. AMD64 and **Intel** 64 systems are supported up to the following limits:

- at most 3 TB memory (theoretical limit: 64 TB)
- at most 160 logical CPUs (theoretical limit: 5120 logical CPUs)

64-bit Power systems now require at least 2 GB of memory to run. They are supported up to the following limits:

- at most 2 TB memory (theoretical limit: 64 TB)
- at most 128 logical CPUs (theoretical limit: 2048 logical CPUs)

IBM System z systems now require at least 1 GB of memory to run, and are theoretically capable of supporting up to the following limits:

- at most 3 TB memory
- at most 101 logical CPUs

The most up to date information about Red Hat Enterprise Linux 7 requirements and limitations is available online at https://access.redhat.com/site/articles/rhel-limits. To check whether your hardware or software is certified, see https://access.redhat.com/certifications.

2.2. INSTALLATION AND BOOT

Read this section for a summary of changes made to installation tools and processes between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

2.2.1. New Boot Loader

Red Hat Enterprise Linux 7 introduces the GRUB2 boot loader, which replaces legacy GRUB in Red Hat Enterprise Linux 7.0 and later. GRUB2 supports more file systems and virtual block devices than its predecessor. It automatically scans for and configures available operating systems. The user interface has also been improved, and users have the option to skip boot loader installation.

However, the move to GRUB2 also removes support for installing the boot loader to a formatted partition on BIOS machines with MBR-style partition tables. This behavior change was made because some file systems have automated optimization features that move parts of the core boot loader image, which could break the GRUB legacy boot loader. With GRUB2, the boot loader is installed in the space available between the partition table and the first partition on BIOS machines with MBR (Master Boot Record) style partition tables. BIOS machines with GPT (GUID Partition Table) style partition tables must create a special BIOS Boot Partition for the boot loader. UEFI machines continue to install the boot loader to the EFI System Partition.

The recommended minimum partition sizes have also changed as a result of the new boot loader. Table 2.1, “Recommended minimum partition sizes” gives a summary of the new recommendations. Further information is available in section 6.14.1.1 MBR and GPT Considerations of the *Red Hat Enterprise Linux 7 Installation Guide*.

Table 2.1. Recommended minimum partition sizes

Partition	BIOS & MBR	BIOS & GPT	UEFI & GPT
/boot	500 MB	/	10 GB
swap	At least twice the RAM. See Section 6.10.4.5. Recommended Partitioning Scheme in the *Red Hat Enterprise Linux 7 Installation Guide* for details.	boot loader	N/A (Installed between the partition table and the first partition)

Users can install GRUB2 to a formatted partition manually with the **force** option at the risk of causing file system damage, or use an alternative boot loader. For a list of alternative boot loaders, see the *Red Hat Enterprise Linux 7 Installation Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

If you have a dual-boot system, use GRUB2’s operating system detection to automatically write a configuration file that can boot either operating system:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

IMPORTANT

Note, that if you have a dual-boot that is based on using UEFI uses other mechanism than MBR legacy based one. This means that you do not need to use EFI specific grub2 command:

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

2.2.1.1. Default Boot Entry for Debugging

Default boot entry for **systemd** has been added to the **/etc/grub.cfg** file. It is no longer necessary to enable debugging manually. The default boot entry allows you to debug systems without affecting options at the boot time.

2.2.2. New Init System

systemd is the system and service manager that replaces the SysV init system used in previous releases of Red Hat Enterprise Linux.

systemd is the first process to start during boot, and the last process to terminate at shutdown. It coordinates the remainder of the boot process and configures the system for the user. Under **systemd**, interdependent programs can load in parallel, making the boot process considerably faster.

systemd is largely compatible with SysV in terms of user experience and scripting APIs. However, some exceptions do exist. See Section 2.2.2.1, “Backwards Compatibility” for details.

The move to **systemd** also involves a change in administration tools for Red Hat Enterprise Linux. See the **systemctl** man page or the *Red Hat Enterprise Linux 7 System Administrator’s Guide* for details.

For further information about the boot process, see the Red Hat Enterprise Linux 7 Installation Guide . For further information about **systemd**, see the Red Hat Enterprise Linux 7 System Administrator’s Guide.

2.2.2.1. Backwards Compatibility

systemd is designed to be largely compatible with SysV in terms of user experience and scripting APIs. However, there are some cases where compatibility is limited.

- Standard **/etc/init.d/*servicename*** commands (**start**, **stop**, **status**) still work. However, Red Hat recommends **/usr/sbin/service *servicename*** commands, as they forward directly to **systemd** rather than using legacy init scripts.

- Run level support is limited. All SysV run levels map to **systemd** targets; however, not all **systemd** targets map to SysV run levels. Some checks for the current run level will therefore return **N** (unknown run level). Red Hat recommends avoiding run level checks and moving to the more useful **systemd** targets.

- Legacy run levels 2, 3, and 4 all map to the **multi-user.target** **systemd** target by default. Users can modify this behavior by configuring different **systemd** targets.

- Services execute cleanly and do not inherit any context of the invoking user. Init scripts depending on inherited context will not work.

- **systemd** does not support additional verbs in init scripts. If you require verbs other than **start**, **stop**, or **status**, move them to an auxiliary script.

- Linux Standard Base header information is now fully interpreted and utilized by **systemd** at run time.

- All init script operations are now subject to a timeout of 5 minutes to prevent the system from freezing because of a hanging init script.

- **systemd** stops only running services; services that were not started are also not stopped during shutdown.

- The **chkconfig** tool shows only SysV services and run level information, and may output misleading information. Red Hat recommends using the **systemctl** command instead.

- SysV services, even those with root privileges, cannot acquire real-time scheduling when the **CPUAccounting** option is enabled. With **CPUAccounting** enabled for any service, **systemd** makes use of the CGroup CPU bandwidth controller globally, and subsequent **sched_setscheduler()** system calls terminate unexpectedly due to real-time scheduling priority. To avoid this error to recur, the CGroup **cpu.rt_runtime_us** option can be set for the real-time using service.

- [illegible] (stdin). If you [illegible] interactive scripts

- Services can no longer read from standart input (stdin). If you require interactive scripts, consider the minimal password querying framework supported by **systemd**. Further information about this functionality is available from the man page:

  ```
  $ man systemd-ask-password
  ```

- Previous versions of Red Hat Enterprise Linux included a System z specific preinstallation script (**linuxrc.s390**), which started System z systems at boot time. The new init system obsoletes this preinstallation script, and System z systems now boot in the same way as AMD64, **Intel** 64 and Power systems.

2.2.2.2. Systemd-debug-generator

The **systemd-debug-generator** is a generator for enabling a runtime debug shell and masking specific units at boot. It reads the kernel command line and understands the following three options:

systemd.mask=

If this option is specified and followed by a unit name, this unit is masked for the runtime. At boot, it is useful to hae certain units removed from the initial boot transaction for debugging system startup. This option may be specified more than once.

systemd.wants=

If this option is specified and followed by a unit name, a start job for this unit is added to the initial transaction. This is useful if you want to start one or more additional units at boot. This option may be specified more than once.

systemd.debug_shell

If this option is specified, the **debug-shell.service** is pulled into the boot transaction. It will spawn a debug shell on tty9 during early system startup. Note that the shell may also be turned on persistently by using a **systemctl enable** command.

2.2.2.3. New Installer

The Red Hat Enterprise Linux installer, Anaconda, has been redesigned and enhanced in order to improve the installation process for Red Hat Enterprise Linux 7.

The updated installer features:

- A redesigned graphical user interface that is faster, more flexible, and requires less input from the user.

- Support for LVM thin provisioning.

- Installation support for btrfs. (Note, however, that btrfs is a Technology Preview in Red Hat Enterprise Linux 7.)

- Improved localization support.

- Support for directly formatted and not partitioned devices.

- Support for teaming and bonding network technologies.

- Support for automatically selecting an appropriate keyboard layout, language, and time zone. (This requires internet connectivity.) Values set based on detection are overridden by any manually set values.

- NTP servers advertised by DHCP are now used automatically.
- Kickstart integration for the **realmd** D-Bus service, Active Directory, and FreeIPA.
- A new text mode that works on IBM System z and PowerPC systems, and serial consoles. Text mode provides a subset of the features provided by the graphical installer.

This new installer also comes with some important changes.

- Previously, storage configuration required that the user have detailed technical knowledge of their storage system. In Red Hat Enterprise Linux 7, storage configuration has been redesigned so that users need to enter minimal detail to configure storage.
- Anaconda now uses the **inst.repo** parameter to set network and other install locations instead of using the **inst.root** parameter.
- Detailed package selection in the graphical installer interface has been replaced by the **Software Selection** screen. Software is divided up in to **Environments** and **Addons**. Users pick one environment and any number of add-ons. Kickstart installations continue to have full control over packages selected at install time.

For further information about any of these features, see the *Red Hat Enterprise Linux 7 Installation Guide*.

2.2.2.4. Boot parameter changes

2.2.2.4.1. Specifying boot parameters

Boot options specific to the installer are prefixed with **inst.** in this guide. Currently, this prefix is optional in Red Hat Enterprise Linux 7: **resolution=1024x768** works exactly the same as **inst.resolution=1024x768**. However, this prefix is expected to become mandatory in future releases, and parameters without prefix are considered deprecated.

2.2.2.4.2. Changes to boot parameters

The new installer uses dracut to configure disks and networking. As a result, some kernel command line boot parameters have changed between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

New parameters

inst.stage2

Specifies the location of the installation program runtime image to be loaded. The syntax is the same as the syntax of the **inst.repo** parameter. This option ignores everything but the image; it cannot be used to specify the location of packages.

inst.dd

Updates a driver package with a package at the location specified. This option can be used multiple times. The location syntax is the same as the location syntax of the **inst.repo** parameter.

inst.geoloc

Configures geolocation usage in the installer to preset the language and time zone. The default value is **provider_fedora_geoip**. Valid values for this parameter include the following.

Table 2.2. Geolocation values

Value	Effect
0	Disables geolocation.
provider_fedora_geoip	Uses the Fedora GeoIP API.
provider_hostip	Uses the Hostip.info GeoIP API.

inst.usefbx

Specifies that the frame buffer X driver should be used instead of a hardware specific driver. This option is equivalent to **inst.xdriver=fbdev**.

bootdev

Specifies the boot interface. This option is mandatory if **ip** is specified more than once.

inst.multilib

Configures the system for multilib packages, for example, to allow 32-bit packages to be installed on a 64-bit system.

gpt

Installs partition information into a GUID Partition Table (GPT) instead of the Master Boot Record (MBR).

inst.virtiolog

Specifies a virtio port to be used to forward logs. The default value is **org.fedoraproject.anaconda.log.0**. If this port exists, it will be used.

rd.dasd

Takes a Direct Access Storage Device (DASD) adaptor device bus identifier and, optionally, comma separated **sysfs** parameter and value pairs. Activates the DASD with the specified device bus ID and sets the mentioned **sysfs** parameters to the values specified. For example, **rd.dasd=adaptor_id,readonly=0**. This parameter can be specified multiple times to activate multiple DASDs.

rd.zfcp

Takes a SCSI over FCP (zFCP) adaptor device bus identifier, a world wide port name (WWPN), and a FCP LUN. Activates the zFCP device with the specified device bus identifier, port name, and LUN. This parameter can be specified multiple times to activate multiple zFCP devices.

```
rd.zfcp=0.0.4000,0x5005076300C213e9,0x5022000000000000
```

rd.znet

Takes a network protocol type, a comma delimited list of subchannels, and, optionally, comma delimited **sysfs** parameter and value pairs. Activates the System z network device driver for the specified protocol, sets up the specified subchannels, and sets the parameters specified. This parameter can be specified multiple times to activate multiple network devices.

```
rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,layer2=1,portname=foo
rd.znet=ctc,0.0.0600,0.0.0601,protocol=bar
```

–

Changed parameters

inst.ks.sendmac

Previously **kssendmac**. Adds headers to outgoing HTTP requests including the MAC addresses of all network interfaces. This is useful when using **inst.ks=http** to provision systems.

name server

Previously **dns**. Specifies the address of the name server. This option can be used multiple times.

Deprecated parameters

Options in this list are deprecated. They will still work, but there are other options which offer the same functionality. Using deprecated options is not recommended and they are expected to be removed in future releases.

updates

Specified the location of updates for the installation program. Use the **inst.updates** option instead.

method

Configured the installation method. Use the **inst.repo=** option instead.

repo

In NFS installations, specified that the target was an ISO image located on an NFS server instead of an installable tree. The difference is now detected automatically, making this option 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.

ip=bootif

Specified the **BOOTIF** option used when installing from a PXE server. This is now detected automatically.

ksdevice

Configured the network device to be used during a Kickstart installation. Different values for this parameter have been replaced with different parameters as per the following table.

Table 2.3. kickstart parameter values

Value	Current behavior
Not present	Attempt to activate all devices with DHCP, unless a device and configuration are specified with the **ip** or **BOOTIF** options.
ksdevice=link	Ignored (this is the same as the default behavior).

Value	Current behavior
ksdevice=bootif	Ignored (**BOOTIF** is used as the default if specified).
ksdevice=ibft	Replaced with **dracut** option **ip=ibft**.
ksdevice=MAC	Replaced with **BOOTIF=MAC**.
ksdevice=device	Replaced with device specification in the **dracut** **ip** option.

blacklist

Used to disable specified drivers. This is now handled by the **rd.driver.blacklist** **dracut** option with the following syntax:

```
rd.driver.blacklist=mod1,mod2,...
```

nofirewire

Disabled support for the FireWire interface. You can disable the FireWire driver (**firewire_ohci**) by using the **rd.driver.blacklist** option instead:

```
rd.driver.blacklist=firewire_ohci
```

Removed parameters

The following options have been removed. They were present in previous releases of Red Hat Enterprise Linux, but they cannot be used anymore.

serial

This option forced Anaconda to use the **/dev/ttyS0** console as the output. Use the **console** parameter to specify the **/dev/ttyS0** console (or similar) instead.

essid, wepkey, wpakey

Configured wireless network access. Network configuration is now 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.

inst.mediacheck

Verified the installation media before starting the installation. Replaced with the **rd.live.check** option.

ks=floppy

Specified a diskette as the Kickstart file source. Floppy drives are no longer a supported boot medium.

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

Disabled 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 dracut** option.

upgradeany

Upgrades have changed inRed Hat Enterprise Linux 7. For more information, see Chapter 1, *How to Upgrade*, Section 3.1.1, "Preupgrade Assistant", and Section 3.1.2, "Red Hat Upgrade Tool" .

vlanid

Configured a VLAN device. Replaced with the **dracut vlan** option.

2.2.3. Changes to firstboot Implementation

Red Hat Enterprise Linux 7 replaces **firstboot** with the Initial Setup utility, **initial-setup**, for better interoperability with the new installer. Basic **firstboot** functionality has been moved to the installer and **initial-setup**.

Third-party modules written for **firstboot** continue to work in Red Hat Enterprise Linux 7. However, **firstboot** is expected to be deprecated in future releases. Maintainers of third-party modules should therefore consider updating their modules for use with the installer or the Initial Setup tool.

2.2.4. Changed mount behavior at boot

Earlier versions of Red Hat Enterprise Linux booted regardless of whether all partitions specified in **/etc/fstab** could be mounted. This could result in a system appearing "up" and healthy, while booting without required partitions.

To prevent this situation, in Red Hat Enterprise Linux 7, if a partition defined in **/etc/fstab** cannot be mounted at boot, boot fails. If a partition should not cause boot to fail in the event that it cannot be mounted, use the new **nofail** parameter in **/etc/fstab**.

```
/dev/critical   /critical  xfs  defaults     1 2
/dev/optional   /optional  xfs  defaults,nofail  1 2
```

In this example, the device mounted at **/optional** would not cause boot to fail if it could not be mounted successfully.

2.2.5. Changes to /etc/issue file

In the previous versions of the Red Hat Enterprise Linux, the **/etc/issue** file contained the product name and the release number of the machine. As of Red Hat Enterprise Linux 7, the product name and the release number have been moved into the **/etc/os-release** file and the first line of **/etc/issue** now contains an **agetty** escape code **\S**. The **\S** escape code expands in the console displaying a product name and the release number of the machine. The code is represented by the **PRETTY_NAME** variable, which is defined in the **/etc/os-release** file.

IMPORTANT

The expansion of the **\S** escape code works only from the console. Using the expansion in an environment that does not support it will result in printing only "\S".

For more information about **\S**, see the **agetty** man pages.

2.3. FILE SYSTEM LAYOUT

Red Hat Enterprise Linux 7 introduces two major changes to the layout of the file system.

- The **/bin**, **/sbin**, **/lib** and **/lib64** directories are now under the **/usr** directory.
- The **/tmp** directory can now be used as a temporary file storage system (**tmpfs**).
- The **/run** directory is now used as a temporary file storage system (**tmpfs**). Applications can now use **/run** the same way they use the **/var/run** directory.

2.3.1. New layout for root file system

Traditionally, only the minimum necessary content was included in the **/bin** and **/lib** directories to avoid slowing down the boot process. Some of the utilities needed to be at the root (/) level in order to mount the **/usr** partition. This created a situation where other utilities spread their content over multiple levels of directories, for example, in both **/bin** and **/usr/bin**.

Red Hat Enterprise Linux 7 moves the **/bin**, **/sbin**, **/lib** and **/lib64** directories into **/usr**. Because the **/usr** file system can now be mounted by **initramfs** rather than by utilities in root level directories, there is no longer a need to split package contents between the two different directory levels. This allows for a much smaller root file system, enabling systems that can more efficiently share disk space, and systems that are easier to maintain, more flexible, and more secure.

To lessen the impact of this change, the previous **/bin** directory is now a symbolic link to **/usr/bin**, **/sbin** to **/usr/sbin**, and so on.

2.3.1.1. Preparing your file system for upgrade

WARNING

Note, that if **/usr** is on a separate partition, the in-place upgrade is not possible. If you decide to move the **/usr** from the separate partition, be aware that this is at your own risk.

If **/var** is on a separate partition, you must manually convert **/var/run** and **/var/lock** to symbolic links:

```
# mv -f /var/run /var/run.runmove~
# ln -sfn /run /var/run
# mv -f /var/lock /var/lock.lockmove~
# ln -sfn /run/lock /var/lock
```

IMPORTANT

You must make sure you address all preupgrade-assistant results regarding partitioning scheme.

When your preparations are complete, see the Red Hat Enterprise Linux 7 Installation Guide for additional details on performing the upgrade process.

2.3.1.2. Verifying a successful upgrade

After performing the upgrade process, it is important to verify that the upgrade worked as expected.

1. Check that the following symbolic links exist:
   - **/bin** is a symbolic link to **/usr/bin**
   - **/sbin** is a symbolic link to **/usr/sbin**
   - **/lib** is a symbolic link to **/usr/lib**
   - **/lib64** is a symbolic link to **/usr/lib64**
   - **/var/run** is a symbolic link to **/run**
   - **/var/lock** is a symbolic link to **/run/lock**
     If the directories listed are symbolic links, as expected, two more checks are required.
2. Check the output of the following find command:

   ```
   # find /usr/{lib,lib64,bin,sbin} -name '.usrmove'
   ```

   Files or directories displayed in response to this command could not be copied to **/usr** because a file or directory with the same name was already present in **/usr**. You will need to manually resolve these naming conflicts.
3. Check the following directories for files that you want to keep:
   - **/var/run.runmove~**
   - **/var/lock.lockmove~**

If any of the directories listed are not symbolic links, you will need to follow the recovery process outlined in Section 2.3.1.3, “Recovering from a failed upgrade” .

2.3.1.3. Recovering from a failed upgrade

The upgrade process may fail for a number of reasons. Check the output of the following commands to see what went wrong:

```
# dmesg
# journalctl -ab --full
```

If no errors are visible, check that:

- / is writable
- /**usr** is writable
- / has sufficient space
- /**usr** has sufficient space
- /**var** is mounted in the **rhelup** tool

Contact Red Hat Support if you need further assistance.

2.3.2. Changes to the /tmp directory

Red Hat Enterprise Linux 7 offers the ability to use /**tmp** as a mount point for a temporary file storage system (**tmpfs**).

When enabled, this temporary storage appears as a mounted file system, but stores its content in volatile memory instead of on a persistent storage device. No files in /**tmp** are stored on the hard drive except when memory is low, in which case swap space is used. This means that the contents of /**tmp** are not persisted across a reboot.

To enable this feature, execute the following command:

```
# systemctl enable tmp.mount
```

To disable this feature, execute the following command:

```
# systemctl disable tmp.mount
```

Red Hat recommends the following uses for the various types of temporary storage space in Red Hat Enterprise Linux 7.

- Privileged processes, such as daemons, should use /**run**/***processname*** to store temporary data.
- Processes that store a large amount of data, or require temporary data to persist across reboots, should use /**var**/**tmp**.
- All other processes should use /**tmp** to store temporary data.

2.3.3. Changes to the /run directory

IMPORTANT

Preupgrade Assistant did not yet check for the effects of this change in the initial release of Red Hat Enterprise Linux 7.0. This issue was corrected in RHBA-2014:1627, available here: https://rhn.redhat.com/errata/RHBA-2014-1627.html.

Previous versions of Red Hat Enterprise Linux allowed some programs to store runtime data in the /**dev** directory during early boot, prior to the /**var** directory being mounted. Consensus between major Linux distributions is that /**run** should be used instead, as the /**dev** directory should be used only for device nodes.

Therefore, in Red Hat Enterprise Linux 7, the **/run** directory is a temporary file storage system (**tmpfs**) that bind mounts the **/var/run** directory. Likewise, the **/run/lock** directory now bind mounts the **/var/lock** directory. Files stored in **/run** and **/run/lock** are no longer persistent and do not survive a reboot. This means that applications must recreate their own files and directories on startup, rather than doing this once at installation time. An **/etc/*app_name*** directory would be ideal for this.

For details on how to recreate files and directories at startup, see the **tmpfiles.d** man page: **man tmpfiles.d**. For example configuration, see the configuration files in **/etc/tmpfiles.d**.

2.4. SYSTEM MANAGEMENT

Read this section for a summary of changes made to system management tools and processes between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

2.4.1. Default process maximums (ulimit)

In Red Hat Enterprise Linux 6, non-root users were restricted to a total of 1024 processes per PAM session. In Red Hat Enterprise Linux 7, this has been increased to 4096 processes per PAM session by default.

The default value is specified in the **/etc/security/limits.d/*-nproc.conf** file (usually **/etc/security/limits.d/20-nproc.conf** on Red Hat Enterprise Linux 7). If this file is not present, the maximum number of processes that a non-root user can own is determined programmatically, as described in https://access.redhat.com/solutions/218383.

You can find out the current number of processes available to non-root users per PAM session by running the **ulimit -u** command.

2.4.2. Configuration File Syntax

In Red Hat Enterprise Linux 6, the **export** command was used in configuration files to export the values defined in those files. Variables that did not use the **export** command were not exported and were used only as configuration values for the corresponding init script. This is an example **/etc/sysconfig/sshd** file:

```
AUTOCREATE_SERVER_KEYS=YES
export SSH_USE_STRONG_RNG=1
export OPENSSL_DISABLE_AES_NI=1
```

In Red Hat Enterprise Linux 6, only the values of **SSH_USE_STRONG_RNG** and **OPENSSL_DISABLE_AES_NI** were exported to the environment of the ssh daemon. The variable **AUTOCREATE_SERVER_KEYS** was used to tell the init script to automatically create RSA and DSA server private and public keys.

In Red Hat Enterprise Linux 7, the **export** command is no longer required for these values to be exported to the environment of the service being configured. Therefore the following example **/etc/sysconfig/sshd** file exports all three values to the environment of the ssh daemon:

```
AUTOCREATE_SERVER_KEYS=YES
SSH_USE_STRONG_RNG=1
OPENSSL_DISABLE_AES_NI=1
```

2.4.3. New Logging Framework

Red Hat Enterprise Linux 7 introduces a new logging daemon, **journald**, as part of the move to **systemd**. **journald** captures the following types of message for all services:

- **syslog** messages
- kernel messages
- initial RAM disk and early boot messages
- messages sent to standard output and standard error output

It then stores these messages in native journal files: structured, indexed binary files that contain useful metadata and are faster and easier to search.

Journal files are not stored persistently by default. The amount of data logged depends on the amount of free memory available; when the system runs out of space in memory or in the **/run/log/journal** directory, the oldest journal files will be removed in order to continue logging.

On Red Hat Enterprise Linux 7, **rsyslog** and **journald** coexist. The data collected by **journald** is forwarded to **rsyslog**, which can perform further processing and store text-based log files. By default, **rsyslog** only stores the journal fields that are typical for **syslog** messages, but can be configured to store all the fields available to **journald**. Red Hat Enterprise Linux 7 therefore remains compatible with applications and system configurations that rely on **rsyslog**.

For further details about the logging subsystem, see the *Red Hat Enterprise Linux 7 System Administrator's Guide*.

2.4.4. Localization Settings

As part of the move to the new init system, **systemd**, localization settings have moved from **/etc/sysconfig/i18n** to **/etc/locale.conf** and **/etc/vconsole.conf**.

2.4.5. Hostname Definition

In Red Hat Enterprise Linux 6, the **hostname** variable was defined in the **/etc/sysconfig/network** configuration file. In Red Hat Enterprise Linux 7, as part of the move to the new init system (**systemd**), the **hostname** variable is defined in **/etc/hostname**.

2.4.6. Updates to Yum

Red Hat Enterprise Linux 7 includes an updated version of **yum**, which includes a number of changes and enhancements. This section lists changes that may affect **yum** users moving from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7.

- **yum group** and **yum groups** are now top level commands, to improve the consistency of command line **yum** use. For example, where previously you would use **yum groupinfo**, you can now use **yum group info**.
- **yum group list** now includes additional optional parameters to alter its output. The new options are **language** and **ids**.
- The default value for the **group_command** parameter in **/etc/yum.conf** has been changed from **compat** to **objects**. Previously, the default behavior of **yum group install** was to install all members of a package group and upgrade both previously installed packages and packages

that had been added to the group since the previous upgrade. The new default behavior is that **yum** keeps track of the previously installed groups and distinguishes between packages installed as a part of the group and packages installed separately.

- The **yum-security** and **yum-presto** plug-ins have been integrated into **yum**.
- **yum** can now download multiple packages simultaneously.
- **yum** now includes support for environment groups. This allows you to install and remove multiple package groups listed under an environment group as a single entity.
- **yum** can now treat a repository as a set of packages, allowing users to treat all packages in a repository as a single entity, for example, to install or remove all packages in that repository. This capability is provided by the **repository-packages** subcommand.
- **yum** now includes a **--disableincludes** option, which allows you to disable **include** statements defined in your configuration files. You can either disable all **include** statements with the **all** value, or disable the **include** statements defined for a specific repository by providing that repository identifier.
- **yum** now includes an **--assumeno** option, which assumes that the answer to any question asked by yum is 'no'. This option overrides the **--assumeyes** option, but is still subject to the behavior prescribed by **alwaysprompt**.

For further information about **yum**, see the man page:

```
$ man yum
```

2.4.7. Updates to RPM Package Manager (RPM)

Red Hat Enterprise Linux 7 provides an updated version of RPM Package Manager. This update includes a number of changes to behavior that may affect migration.

- Conflict detection is now stricter and more correct. Some packages that would have installed on Red Hat Enterprise Linux 6 may not install on Red Hat Enterprise Linux 7 because of this heightened conflict sensitivity.
- A package that conflicts with other versions of itself can now be set up as a singleton using alternatives, so that multiple versions of a single package can be installed alongside each other.
- If an installed package lists another package as obsolete, the second package is not installed.
- Obsolete rules now include all matching packages regardless of other attributes such as architecture.
- Dependency calculations no longer consider files that were not installed or files that were replaced, for example, with the **--nodocs**, **--noconfig**, or **--force** options, as being provided.
- There is no longer a need to manually execute **rm -f /var/lib/rpm/__db.** when rebuilding a panicked (**DB_RUNRECOVER**) RPM Package Manager database.
- Public keys created with OpenPGP 3 are no longer supported.
- The **--info** option now outputs individual tag–value pairs per line to improve human readability. Any scripts that rely on the previous **--info** format need to be rewritten.

- The spec parser is now stricter and more correct, so some previously-accepted spec files may fail to parse, or give warnings.
- **%license** can now be used to mark files in the **%files** section of a spec file as licenses that must be installed even when **--nodocs** is specified.
- Version comparison now supports the dpkg-style tilde (~) operator to handle pre-release software better. For example, **foo-2.0~beta1** is considered older than **foo-2.0**, removing the need for tricks with the Release field to handle these common upstream version practices.
- The automatic dependency generator has been rewritten into an extensible, customizable rule-based system with built in filtering.

This update also includes the following enhancements:

- It is now possible to query the files installed from a package (**INSTFILENAMES**), the number of hard links to a file (**FILENLINKS**), package version control system details (**VCS**), and formatted dependency string shortcuts (**PROVIDENEVRS**, **REQUIRENEVRS**, **CONFLICTNEVRS**, **OBSOLETENEVRS**).
- A number of new commands are provided, including:
  - **rpmkeys**
  - **rpmdb**
  - **rpmspec**
  - **rpmsign**
- RPM Package Manager now includes new switches to scriptlets to enable runtime macro expansion or runtime query format expansion.
- Pre- and post-transaction scriptlet dependencies can now be correctly expressed with **Requires(pretrans)** and **Requires(posttrans)**.
- RPM Package Manager now includes the **OrderWithRequires** tag to allow users to supply additional ordering information. This new tag uses the same syntax as the Requires tag, but does not generate dependencies. If mentioned packages are present in the same transaction, the ordering hints are treated like **Requires** when calculating transaction order.
- Line continuations and macro expansions in spec files are no longer limited to a specified length.
- RPM Package Manager now allows users to specify upstream version control repository information.
- RPM Package Manager now includes an **%autosetup** macro to assist in automating the process of applying patches.

2.4.8. New Format of ifconfig

The format of output from the deprecated **ifconfig** tool has changed in Red Hat Enterprise Linux 7. Scripts that parse **ifconfig** output may be affected by these changes, and may need to be rewritten.

Red Hat recommends using the **ip** utility and its subcommands (**ip addr**, **ip link**) instead of the deprecated **ifconfig** tool.

2.4.9. Changes to Control Groups

The kernel uses control groups to group processes for the purpose of system resource management. Red Hat Enterprise Linux 7 introduces a number of changes to control groups.

- Control groups are now mounted under **/sys/fs/cgroup** instead of **/cgroup**.
- Some file systems are now mounted by default.
- **systemd** does not yet fully support migration from **libcgroup** to **systemd**. As such, the **cgred** service should be used only to move processes to groups not managed by **systemd**. The **cgconfig.conf** file should be used to configure a control group hierarchy for file systems or file controllers not managed by **systemd**.

For further information about these changes, see the *Red Hat Enterprise Linux 7 Resource Management Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.4.10. Changes to Kernel Crash Collection (Kdump)

The kernel crash collection tool, **kdump**, previously generated an initial RAMDisk (**initrd**) for the **kdump** capture kernel with a custom **mkdumprd** script. In Red Hat Enterprise Linux 7 the initial RAMDisk is generated with dracut, making the process of generating the initial RAMDisk easier to maintain.

As a result of this move, the following changes have been made to **kdump** and its configuration files.

- The **net** directive is no longer supported. Users must now explicitly define either **ssh** or **nfs**.
- The **blacklist** option is no longer supported. Instead, users can specify **rd.driver.blacklist** as a parameter in the **/etc/sysconfig/kdump** file of their capture kernel.
- The default **mount_root_run_init** action, which was performed if dumping to an intended target failed, has been replaced by the **dump_to_rootfs** action. Instead of mounting the real root file system, running init scripts, and attempting to save the vmcore when the **kdump** service has started, this new action mounts the root file system and saves the vmcore to it immediately.
- A new directive, **dracut_args**, allows you to specify additional dracut arguments when configuring kdump.
- The **debug_mem_level** option is no longer included in **kdump**. This functionality has been moved to dracut. Users can achieve the same functionality by specifying **rd.memdebug** as a parameter in the **/etc/sysconfig/kump** file of their capture kernel.
- The **options** directive was previously used to include parameters specific to the kernel module in the initial ram file system (**initramfs**). This method is not supported in Red Hat Enterprise Linux 7. Instead, users can specify relevant parameters in the **/etc/sysconfig/kdump** file of their capture kernel.
- The **link_delay** and **disk_timeout** parameters are no longer necessary or supported, as dracut contains **udev**, which addresses the use case for which these parameters were previously required.
- Any file system back-end dump targets must be mounted in the crashed kernel before the **kdump** service is started and the initial RAMDdisk image is created. You can achieve this by adding these targets to **/etc/fstab** to be automatically mounted at boot time.
- If you specify a path, but do not specify a target, and any directory in the path that you specify is a mount point for a separate device, the vmcore is saved to the path, not the device mounted

somewhere along that path. Therefore when your system reboots, and the device mounts, the vmcore is inaccessible, because the device has mounted over the top of its location. Red Hat Enterprise Linux 7 now warns about this issue when you specify a path without specifying a target.

For further details about **kdump**, see the *Red Hat Enterprise Linux 7 Kernel Crash Dump Guide* , available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/

2.4.11. Changes to usermod behavior

In Red Hat Enterprise Linux 6, the **-g** option of the **usermod** command did not manipulate group ownership. From Red Hat Enterprise Linux 7.0 to Red Hat Enterprise Linux 7.2 release, the **-g** option modified the group ownership of the files in the **/home** directory tree. Starting from Red Hat Enterprise Linux 7.3, **usermod** changes the group ownership of the files inside of the user's home directory only if the home directory user ID matches the user ID being modified.

2.4.12. Changes to System accounts

The default range of IDs for system users, normal users and groups has changed in Red Hat Enterprise Linux 7 release as follows:

Table 2.4. ID layout

Range	Red Hat Enterprise Linux 6	Red Hat Enterprise Linux 7
System accounts	0-499	0-999
User accounts	500-60,000	1,000-60,000

This change might cause problems when migrating to Red Hat Enterprise Linux 7 with existing users having UIDs and GIDs between 500 and 999. The default ranges of UID and GID can be manually changed in the **/etc/login.defs** file.

2.5. FILE SYSTEM FORMATS

Read this section for a summary of changes to file system format support between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

2.5.1. New Default File System: XFS

XFS is a very high performance, scalable file system and is routinely deployed in the most demanding applications. In Red Hat Enterprise Linux 7, XFS is the default file system and is supported on all architectures.

Ext4, which does not scale to the same size as XFS, is fully supported on all architectures and will continue to see active development and support.

Details of Red Hat support limits for XFS are available at https://access.redhat.com/site/articles/rhel-limits.

For further details about using and administering the XFS file system, see the *Red Hat Enterprise Linux 7 Storage Administration Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.5.1.1. Changes to mount options

Unlike ext3 and ext4, the XFS file system enables the **user_xattr** and **acl** mount options by default. This means that you will encounter errors like the following if you include these options at either the command line or in **/etc/fstab**.

```
$ mount -o acl /dev/loop0 test
mount: wrong fs type, bad option, bad superblock on /dev/loop0,
    missing codepage or helper program, or other error

    In some cases useful info is found in syslog - try
    dmesg | tail or so.
```

Ext3 and ext4 file systems do not enable these attributes by default, and accept these options when you use the **mount** command or mount them with **/etc/fstab**.

2.5.2. Btrfs Technology Preview

Red Hat Enterprise Linux 7 introduces btrfs as a Technology Preview. Btrfs is a next generation Linux file system that offers advanced management, reliability, and scalability features. Btrfs provides checksum verification for files as well as metadata. It also offers snapshot and compression capabilities, and integrated device management.

Details of Red Hat support limits for btrfs are available at https://access.redhat.com/site/articles/rhel-limits. For more information about the level of support available for Technology Preview features, see https://access.redhat.com/site/support/offerings/techpreview/.

For further details about using and administering btrfs, see the *Red Hat Enterprise Linux 7 Storage Administration Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.5.2.1. Kickstarting btrfs

In a kickstart file, to create a partition on the system, you would usually use the **part** command with the **--fstype** to create a partition that used a particular file system, like so:

```
part /mnt/example --fstype=xfs
```

However, in Red Hat Enterprise Linux 7.0 and 7.1, btrfs is treated more as a device type than a file system type. As such, **btrfs** is not a valid value for the **--fstype** parameter. Instead, use the **btrfs** command to define a btrfs volume, like so:

```
btrfs mount_point --data=level --metadata=level --label=label partitions
```

2.5.3. Extended file system support

Red Hat Enterprise Linux 7 introduces a unified extended file system driver that provides support for Ext2, Ext3, and Ext4.

However, Ext2 is considered deprecated as of Red Hat Enterprise Linux 7, and should be avoided if possible.

For further information about these file systems, see the *Red Hat Enterprise Linux 7 Storage Administration Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.6. PHYSICAL STORAGE

Read this section for a summary of changes to support for physical storage and relevant configuration tools between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

2.6.1. Changed mount behavior at boot

If a storage device is configured to mount at boot time, and that device cannot be found, or does not mount correctly, Red Hat Enterprise Linux 7 fails to boot. This is an intentional change of behavior to prevent systems from booting without important storage devices. Earlier versions of Red Hat Enterprise Linux booted regardless of whether all storage devices that were configured to mount at boot were found or mounted correctly.

If a device should not prevent the system from booting, you can mark it with the **nofail** option, as shown.

```
/dev/essential-disk   /essential   xfs auto,defaults    0 0
/dev/non-essential-disk  /non-essential  xfs auto,defaults,nofail  0 0
```

2.6.2. Using LVM snapshots as a rollback mechanism

WARNING

LVM snapshots are not recommended as a primary rollback method. During an upgrade, the entire system (except user files) is overwritten. A snapshot of the system is therefore nearly the same size as the original data set.

Additionally, snapshots are more prone to error than the typical backup process, as they do not include the **/boot** partition.

When upgrading from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7, Red Hat recommends taking a full backup and using the backup as the primary rollback method. LVM snapshots should be used as a secondary rollback method only.

As of Red Hat Enterprise Linux 6.3, users can reserve space on their logical volumes to use as storage space for snapshots. The system can then be rolled back to the snapshot in the event that an upgrade or migration fails.

If you want to use an LVM snapshot as a secondary rollback method, you may need to add space to allow room for a complete snapshot. To add more space, you can do any of the following:

- Add another disk. Instructions can be found in the *Red Hat Enterprise Linux 7 Storage Administration Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

- Use **parted** to check for free space that is not allocated to an existing partition.

- Use **lsblk** to check for empty partitions, or partitions that can be deleted to free space.

- Use **vgdisplay** to check for free space in a volume group that is not allocated to a logical volume.

- Use **df** to check for file systems that have free space and can be reduced, so that their logical volume or partition can be shrunk to free space.

Be aware of the following potential limitations of using LVM snapshots for rollback:

- Snapshot size is not adjusted automatically. If your snapshot gets too large for its partition, it may become invalid, and rollback will fail. It is therefore imperative to allocate a sufficiently large space for a snapshot of your entire system, before creating that snapshot. If you need to resize a root snapshot, you will need an additional device such as a Live CD that can be used as a root device while your original root device is unmounted and resized.

- The copy-on-write device of a snapshot is not mirrored, and will be on a single device regardless of whether your system is mirrored. If the device fails and you lose the snapshot, rollback is impossible. Red Hat recommends using a physical volume with mdraid, or using multiple snapshots to separate disks. Using multiple snapshots is slower.

- In the event of a crash during installation, the system can become impossible to boot. In this circumstance, Red Hat recommends booting with a Live CD or PXE boot and merging your snapshot when the system has booted successfully. Merging instructions are available in the Red Hat Enterprise Linux 7 LVM documentation, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

- Rollback returns **/var/log** to the state it was in prior to upgrade. For auditing purposes, Red Hat recommends copying log files from installation to a separate location prior to initiating rollback.

2.6.3. Target Management with targetcli

Previous versions of Red Hat Enterprise Linux used **tgtd** for iSCSI target support and LIO, the Linux kernel target, only for Fibre-Channel over Ethernet (FCoE) targets through the **fcoe-target-utils** package.

Red Hat Enterprise Linux 7 now uses the LIO kernel target subsystem for FCoE, iSCSI, iSER (Mellanox InfiniBand) and SRP (Mellanox InfiniBand) storage fabrics. All fabrics can now be managed with the **targetcli** tool.

2.6.4. Persistent Device Names

Red Hat Enterprise Linux 7 makes the management of devices on the system easier by storing the mapping of device names (for example, sda, sdb, and others) and persistent device names (provided by **udev** in **/dev/disk/by-*/**) in kernel messages. This lets the system administrator identify the messages associated with a device, even if the device name changes from boot-to-boot.

The kernel **/dev/kmsg** log, which can be displayed with the **dmesg** command, now shows the messages for the symbolic links, which **udev** has created for kernel devices. These messages are displayed in the following format: **udev-alias: *device_name* (*symbolic_link symbolic link* ...)**. For example:

```
udev-alias: sdb (disk/by-id/ata-QEMU_HARDDISK_QM00001)
```

Any log analyzer can display these messages, which are also saved in **/var/log/messages** through **syslog**.

To enable this feature add **udev.alias=1** to the kernel command line in **/etc/default/grub**.

2.6.5. LVM cache volumes

LVM cache volume functionality is fully supported as of Red Hat Enterprise Linux 7.1. This feature allows users to create logical volumes with a small, fast device performing as a cache for larger, slower devices. See the **lvmcache** manual page for information on creating cache logical volumes.

2.7. NETWORKING

Read this section for a summary of changes to networking, network protocol support and relevant configuration tools between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

2.7.1. Recommended naming practices

A host name can be a free-form string of up to 64 characters in length. However, Red Hat recommends that both static and transient names match the fully-qualified domain name (FQDN) used for the machine in DNS, such as **host.example.com**. The **hostnamectl** tool allows static and transient host names of up to 64 characters including a-z, A-Z, 0-9, **-**, and **.** only. Underscores are technically permissible in the current specification. However, since older specifications forbid them, Red Hat does not recommend using underscores in host names.

The Internet Corporation for Assigned Names and Numbers (ICANN) sometimes adds previously unregistered Top-Level Domains (such as **.yourcompany**) to the public register. Therefore, Red Hat strongly recommends that you do not use a domain name that is not delegated to you, even on a private network, as this can result in a domain name that resolves differently depending on network configuration. As a result, network resources can become unavailable. Using domain names that are not delegated to you also makes DNSSEC more difficult to deploy and maintain, as domain name collisions add manual configuration penalties to DNSSEC validation.

For further information about this issue, see the ICANN FAQ on domain name collision: http://www.icann.org/en/help/name-collision/faqs

2.7.2. Updates to NetworkManager

Red Hat Enterprise Linux 7 includes an updated version of **NetworkManager**, which provides a number of enhancements and some new features.

- The **nmcli** tool now supports editing connections with the **nmcli con edit** and **nmcli con modify** commands.

- A new text-based user interface (**nmtui**) provides a streamlined console-based tool for editing network configuration and controlling network connections. This replaces the **system-config-network-tui** tool.

- Previously, **NetworkManager** ignored network interfaces it did not recognize (interfaces other than Ethernet, Infiniband, WiFi, Bridge, Bond, and VLAN). **NetworkManager** now recognizes any network interface picked up by **ip link**, and exposes these interfaces through the D-Bus interface and clients such as **nmcli**. This brings **NetworkManager** to closer parity with tools like **ip**.

- **NetworkManager** now non-destructively takes ownership of interfaces that it can natively configure, such as Ethernet, InfiniBand, Bridge, Bond, VLAN, and Team interfaces. If these interfaces are configured before **NetworkManager** starts or restarts, the previously configured connections are not interrupted. This means that the **NM_CONTROLLED** option is no longer required.

- Support for checking network connectivity, hotspots and portals. This behavior is disabled by default.
- Support for team interfaces.
- Basic, non-native support for GRE, macvlan, macvtap, tun, tap, veth, and vxlan devices.
- A new **NetworkManager-config-server** package provides defaults that are suitable for servers, such as ignoring carrier changes and not creating default DHCP connections.
- A new **dns=none** configuration option for **NetworkManager.conf** prevents **NetworkManager** from making changes to the **resolv.conf** file.
- Support for fast user switching.
- Support for locking a connection to the name of an interface in addition to, or instead of, the MAC address of an interface.

This update also changes configuration file monitoring behavior. **NetworkManager** no longer monitors on-disk configuration files for changes. Instead, users must manually reload changed configuration files with the **nmcli con reload** command.

2.7.3. New Network Naming Schema

Red Hat Enterprise Linux 7 provides methods for consistent and predictable network device naming for network interfaces. These features change the name of network interfaces on a system in order to make locating and differentiating the interfaces easier.

Traditionally, network interfaces in Linux are enumerated as **eth[0123…]**, but these names do not necessarily correspond to actual labels on the chassis. Modern server platforms with multiple network adapters can encounter non-deterministic and counter-intuitive naming of these interfaces. This affects both network adapters embedded on the motherboard (Lan-on-Motherboard, or LOM) and add-in (single and multi-port) adapters.

In Red Hat Enterprise Linux 7, **systemd** and **udevd** support a number of different naming schemes. The default behavior is to assign fixed names based on firmware, topology, and location information. This has the advantage of names that are fully automatic and fully predictable, stay fixed even if hardware is added or removed (no re-enumeration takes place), and that broken hardware can be replaced seamlessly. The disadvantage to this behavior is that the names are sometimes harder to read than the name that has previously been used, for example, **enp5s0** in place of **eth0**.

The following naming schemes for network interfaces are now supported by **udevd** natively.

Scheme 1

Names incorporating Firmware or BIOS provided index numbers for on-board devices, for example, **eno1**. **systemd** names interfaces according to this scheme by default if that information from the firmware is applicable and available, with scheme 2 used as a fallback.

Scheme 2

Names incorporating Firmware or BIOS provided PCI Express hotplug slot index numbers, for example, **ens1**. **systemd** names interfaces according to this scheme by default if that information from the firmware is applicable and available, with scheme 3 used as a fallback.

Scheme 3

Names incorporating physical location of the connector of the hardware, for example, **enp2s0**. **systemd** names interfaces according to this scheme by default if that information from the firmware is applicable and available, with scheme 5 used as a fallback.

Scheme 4

Names incorporating the interface's MAC address, for example, **enx78e7d1ea46da**. By default, **systemd** does not name interfaces according to this scheme, but it can be enabled if required.

Scheme 5

The traditional unpredictable kernel-native ethX naming, for example, **eth0**. **systemd** names interfaces according to this scheme if all other methods fail.

If the system has **BIOSDEVNAME** enabled, or if the user has added **udevd** rules that change the names of kernel devices, these rules will take precedence over the default **systemd** policy.

For further information about this new naming system, see the *Red Hat Enterprise Linux 7 Networking Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.7.4. New networking utility (ncat)

A new networking utility, **ncat**, replaces **netcat** in Red Hat Enterprise Linux 7. **ncat** is a reliable back-end tool that provides network connectivity to other applications and users. It reads and writes data across the network from the command line, and uses both TCP and UDP for communication.

Some of the commands in **ncat** differ from those originally provided by **netcat**, or provide different functionality with the same options. These differences are outlined in the following list.

- The **netcat -P** option took a specified user name to present to a proxy server that required authentication. The **ncat** option for this behavior is **--proxy-auth *user*[:*pass*]**.
- The **netcat -X** option took a specified protocol for the networking utility to use when communicating with a proxy server. The **ncat** option for this behavior is **--proxy-type**.
- The **netcat -x** option took an address and an optional port for the networking utility to connect to with the proxy server. The **ncat** option for this behavior is **--proxy**, which takes an IP address and an optional port, like so: **--proxy *host*[:*port*]**.
- The **netcat -d** option disabled reading from stdin. The **ncat -d** option allows the user to specify a wait time between read or write operations. However, **ncat** provides the **--recv-only** option, which provides similar behavior to **netcat -d**.
- The **netcat -i** option specified an interval between lines of text sent and received, or between connections to multiple ports. The **ncat -i** option specifies the amount of time a connection can idle before the connection times out and is terminated. There is no equivalent in **ncat** to the **netcat -i** option.
- The **netcat -w** option specifies the amount of time a connection that cannot be established can idle before the connection times out and is terminated. The **ncat -w** option specifies the amount of time to attempt connection before timing out.

Some options that were available in **netcat** do not have equivalents in **ncat**. **ncat** cannot currently perform the following.

- Enable debugging on the socket (previously provided by **netcat -D**).

- Specify the size of the TCP send and receive buffers (previously provided by **netcat -I** and **netcat -O**).
- Specify that source or destination ports are chosen randomly (previously provided by **netcat -r**).
- Enable Protection of BGP Sessions avia the TCP MD5 Signature Option, RFC 2385 (previously provided by **netcat -S**).
- Specify the IPv4 type of service (previously provided by **netcat -T**).
- Specify the use of UNIX domain sockets (previously provided by **netcat -U**).
- Specify the routing table to be used (previously provided by **netcat -V**).
- Scan for listening daemons without transmitting data.
- Specify an interval between lines of text sent and received, or between connections to multiple ports.

The **ncat** utility is provided by the **nmap-ncat** package. For more information about **ncat**, see the man page:

```
$ man ncat
```

2.7.5. Changes to Postfix

Red Hat Enterprise Linux 7 upgrades **postfix** from version 2.6 to version 2.10. While major compatibility issues are handled by the Preupgrade Assistant on upgrading from Red Hat Enterprise Linux 6 to 7, users should be aware of the following non-fatal compatibility issues.

- Ensure that you execute **postfix stop** and **postfix start** commands before using the **postscreen** daemon, to avoid problems with the **pass** master service.
- Default system-supplied CA certificates are no longer added to the ***_tls_CAfile** or ***_tls_CApath** lists. This means third-party certificates no longer receive mail relay permission when **permit_tls_all_clientcerts** is used. If your configuration requires certificate verification, enable backwards compatible behavior by setting **tls_append_default_CA = yes**.
- The **verify** service now uses a persistent cache with periodic cleanup enabled by default. Support for the delete and sequence operations is required. To disable the cache, specify a blank **address_verify_map** parameter in **main.cf**. To disable periodic cleanup, set **address_verify_cache_cleanup_interval** to **0**.
- Previously the default next-hop destination, used when a filter next-hop destination was not specified, was the value of **$myhostname**. The default is now the recipient domain. To change the default next-hop destination, specify **default_filter_nexthop = $myhostname**. In pipe-based filters, this also enables FIFO delivery order, instead of round-robin domain selection.
- The **postmulti -e destroy** command no longer attempts to remove files that are created after the **postmulti -e create** command is executed.
- Postfix now requests default delivery status notifications when adding a recipient with the Milter **smfi_addrcpt** action.

- When the result of virtual alias expansion exceeds virtual alias recursion or expansion limits, Postfix now reports a temporary delivery error instead of silently dropping excess recipients and delivering the message.

- The local delivery agent now keeps the owner-alias attribute of a parent alias when delivering mail to a child alias that does not have an owner-alias. This makes repeated delivery to mailing lists less likely. To enable older behavior, specify **reset_owner_alias = yes**.

- The Postfix SMTP client no longer appends the local domain when looking up a DNS name without "**.**". To enable older behavior, specify **smtp_dns_resolver_options = res_defnames**. Note that this may produce unexpected results.

- The format of the **postfix/smtpd[pid]: queueid: client=host[addr]** log file record has changed. When available, the before-filter client information and before-filter queue ID are now appended to the end of the record.

- By default, postfix no longer adds an undisclosed recipient header to messages with no specified recipient. To enable older behavior, specify the following in **mail.cf**:

  ```
  undisclosed_recipients_header = To: undisclosed-recipients:;
  ```

- The SASL mechanism list is now re-computed after each successful completion of **STARTTLS**.

- The **smtpd_starttls_timeout** default value is now stress-dependent.

- DNSBL queries with a secret in the domain name must now hide that secret from **postscreen** SMTP replies. For example, in **main.cf**, specify:

  ```
  postscreen_dnsbl_reply_map = texthash:/etc/postfix/dnsbl_reply
  ```

  In **dnsbl_reply**, specify a separate DNSBL name:

  ```
  # Secret DNSBL name    Name in postscreen(8) replies
  secret.zen.spamhaus.org  zen.spamhaus.org
  ```

- All programs that use postfix VSTREAMs must be recompiled, because VSTREAM errors now use separate flags for read and write errors.

- The default value of **smtp_line_length_limit** is now **999**, to remain consistent with the SMTP standard.

- Sendmail now transforms all input lines ending in **<CR><LF>** into UNIX format (**<LF>**).

- By default, the SMTP client no longer appends **AUTH=<>** to the **MAIL FROM** command.

- Some log messages that were previously classified as **fatal** are now classified as **error**. Log file based alert systems may need to be updated accordingly. To re-enable older behavior, set **daemon_table_open_error_is_fatal** to **yes**.

- Newly supported long queue file names are not supported prior to Postfix 2.9. To migrate back to Postfix 2.8 or earlier, any long queue file names must be converted. To do so, stop postfix, set **enable_long_queue_ids** to **no**, and then run the **postsuper** command until it no longer exports queue file name changes.

- Postfix now logs the result of successful TLS negotiation with TLS logging levels of 0. See log level descriptions in the **postconf** man page for details.

- The postfix SMTP server now always checks the smtpd_sender_login_maps table.
- The default **inet_protocols** value is now **all** (use both IPv4 and IPv6). To avoid unexpected performance loss for sites without global IPv6 connectivity, the **make upgrade** and **postfix upgrade-configuration** commands currently append **inet_protocols = ipv4** to **main.cf** when no explicit setting is present.
- The default **smtp_address_preference** value is now **any** (choose IPv4 or IPv6 at random).
- The SMTP server no longer reports transcripts of sessions where a client command is rejected because a lookup table is not available. To continue receiving such reports, add the **data** class to the value of the **notify_classes** parameter.
- A new **smtpd_relay_restrictions** parameter has been added. By default this enables **permit_mynetworks**, **permit_sasl_authenticated**, and **defer_unauth_destination**. This prevents open relay problems due to mistakes with spam filter rules in **smtpd_recipient_restrictions**. However, if your site has a complex mail relay policy configured under **smtpd_recipient_restrictions**, some mail may be incorrectly deferred. To correct this, either remove **smtpd_relay_restrictions** configuration and use the existing policy in **smtpd_recipient_restrictions**, or copy the existing policy from **smtpd_recipient_restrictions** to **smtpd_relay_restrictions**.

2.7.6. Network Protocols

Read this section for a summary of changes to network protocols between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

2.7.6.1. Network File System (NFS)

Red Hat Enterprise Linux 7 provides support for NFS 3, NFS 4.0, and NFS 4.1. NFS 2 is no longer supported as of Red Hat Enterprise Linux 7.

NFS 4.1 provides a number of performance and security enhancements, including client support for Parallel NFS (pNFS). Additionally, a separate TCP connection is no longer required for callbacks, allowing an NFS server to grant delegations even when it cannot contact the client, for example, when NAT or a firewall interferes.

NFS 3, NFS 4.0, and NFS 4.1 are supported on the server. Support for a particular version can be enabled or disabled in the **/etc/sysconfig/nfs** file, by changing the value of the **RPCNFSDARGS** parameter. For example, **RPCNFSDARGS="-N4.1 -V3"** enables support for NFS 3 and disables support for NFS 4.1. For further details, see the man page:

```
$ man rpc.nfsd
```

NFS clients attempt to mount using NFS 4.0 by default, and fall back to NFS 3 if the mount operation is not successful. Default behavior can be altered by editing the **/etc/nfsmount.conf** file and by using command line options. See the man pages for further details.

```
$ man nfs
```

```
$ man nfsmount.conf
```

2.7.6.1.1. Parallel NFS (pNFS)

Red Hat Enterprise Linux 7 provides client support for Parallel NFS (pNFS). pNFS improves the scalability of NFS and has the potential to improve performance. When the Red Hat Enterprise Linux 7 client mounts a server that supports pNFS, that client can access data through multiple servers concurrently. Note that Red Hat Enterprise Linux 7 supports the files layout type, with objects and blocks layout types being included as a technology preview. For more information about this protocol and its capabilities, see the Red Hat Enterprise Linux 7 Storage Administration Guide .

2.7.6.2. Apache Web Server (httpd)

Red Hat Enterprise Linux 7 provides an updated version of Apache Web Server. This new version (2.4) includes some significant packaging changes as well as a number of new features.

Changed proxy configuration

Apache Web Server (**httpd**) configurations that use an SSL back end must now use the **SSLProxyCheckPeerName** directive if the SSL certificate does not match the host name configured. Previously, host names in the SSL certificate of a proxy back end were not verified.

New control mechanisms

Because Red Hat Enterprise Linux moves the system away from SysV init scripts, the commands for controlling the **httpd** service have changed. Red Hat now recommends the **apachectl** and **systemctl** commands instead of the **service** command. For example, where you would previously have run **service httpd graceful**, Red Hat now recommends **apachectl graceful**.

Changed default subcommand behavior

The **systemd** unit file for **httpd** defines different behavior for the **reload** and **stop** subcommands. Specifically, the **reload** subcommand now gracefully reloads the service, and the **stop** command now gracefully stops the service by default.

Hard coded default configuration

Previous versions of **httpd** provided an exhaustive configuration file that listed all configuration settings and their defaults. Many common configuration settings are no longer explicitly configured in the default configuration files; instead, default settings are now hard coded. The default configuration file now has minimal content and is easier to manage as a result. The hard coded default values for all settings are specified in the manual, which by default is installed into **/usr/share/httpd**.

New Multi-Processing Model modules

Previous releases of Red Hat Enterprise Linux provided several Multi-Processing Models (**prefork** and **worker**) as different **httpd** binaries. Red Hat Enterprise Linux 7 uses a single binary and provides these Multi-Processing Models as loadable modules: **worker**, **prefork** (default), and **event**. Edit the **/etc/httpd/conf.modules.d/00-mpm.conf** file to select which module is loaded.

Directory changes

A number of directories have moved or are no longer provided in this updated version of **httpd**.

- Content previously installed in **/var/cache/mod_proxy** has moved to **/var/cache/httpd** under either the **proxy** or the **ssl** subdirectory.
- Content previously installed in **/var/www** has moved to **/usr/share/httpd**.
- Content previously installed in **/var/www/icons** has moved to **/usr/share/httpd/icons**. This directory contains a set of icons used with directory indices.
- The HTML version of the **httpd** manual previously installed in **/var/www/manual** has moved to **/usr/share/httpd/manual**.

- Custom multi-language HTTP error pages previously installed in **/var/www/error** have moved to **/usr/share/httpd/error**.

Changes to suexec

The **suexec** binary no longer has its user identifier set to root at install time. Instead, a more restrictive set of permissions is applied using file system capability bits. This improves the security of the **httpd** service. Additionally, **suexec** now sends log messages to **syslog** instead of using the **/var/log/httpd/suexec.log** file. The messages sent to **syslog** appear in **/var/log/secure** by default.

Changes to module interface compatibility

Changes to the **httpd** module interface mean that this updated version of **httpd** is not compatible with third-party binary modules built against the previous version of **httpd** (2.2). Such modules will need to be adjusted as necessary for the **httpd** 2.4 module interface, and then rebuilt. See the Apache documentation for details of the API changes in version 2.4.

Change to apxs binary location

The **apxs** binary used to build modules from source has moved from **/usr/sbin/apxs** to **/usr/bin/apxs**.

New and moved configuration files

Configuration files that load modules are now placed in the **/etc/httpd/conf.modules.d** directory. Packages that provide additional loadable modules for **httpd** (like the **php** package) add files to this directory. Any configuration files in the **conf.modules.d** directory are processed before the main body of **httpd.conf**. Configuration files in the **/etc/httpd/conf.d** directory are now processed after the main body of **httpd.conf**.
Some additional configuration files are provided by the **httpd** package:

- **/etc/httpd/conf.d/autoindex.conf** configures **mod_autoindex** directory indexing.

- **/etc/httpd/conf.d/userdir.conf** configures access to user directories (http://example.com/~username/). By default this access is disabled for security reasons.

- **/etc/httpd/conf.d/welcome.conf** configures the "welcome page" displayed on http://localhost/ when no content is present.

Changes to configuration compatibility

This version of **httpd** is not compatible with the configuration syntax of the previous version (2.2). Configuration files require updates to syntax before they can be used with this updated version of **httpd**. See the Apache documentation for details of the syntax changes made between version 2.2 and version 2.4.

2.7.6.3. Samba

Red Hat Enterprise Linux 7 provides Samba 4, a combined set of daemons, client utilities, and Python bindings that allow communicating using SMB1, SMB2, and SMB3 protocols.

The current implementation of Kerberos does not support the Samba 4 Active Directory Domain Controller functionality. This functionality has been omitted from Red Hat Enterprise Linux 7.0, but is expected to be included in future releases. All other functionality that does not rely on the Active Directory DC is included.

Red Hat Enterprise Linux 6.4 and later provided Samba 4 as a Technology Preview, and packaged it as a series of [package]*samba4- **packages to avoid conflicting with the stable Samba 3 packages**

([package]*samba-). Since Samba 4 is now fully supported and provides a number of enhancements over Samba 3, Red Hat Enterprise Linux 7 provides Samba 4 as the standard [package]*samba-**packages. The special [package]*samba4-** packages are obsolete.

For more information about Samba, see the *Red Hat Enterprise Linux 7 System Administrator's Guide* and *System Administrators Reference Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.7.6.4. BIND

In Red Hat Enterprise Linux 6, installing the **bind-chroot** package changed the **ROOTDIR** environment variable in **/etc/sysconfig/named** to point to the chroot environment location. To run the **named** service normally (not in the chroot environment) required either removing the **bind-chroot** package or manually editing the **ROOTDIR** environment variable in **/etc/sysconfig/named** file.

In Red Hat Enterprise Linux 7, installing the **bind-chroot** package does not change how the **named** service runs. Instead, it installs a new service, **named-chroot**, that is started and stopped separately with the **systemctl** command, like so.

```
# systemctl start named-chroot.service
```

```
# systemctl stop named-chroot.service
```

The **named-chroot** service cannot run at the same time as the **named** service.

2.7.7. Default product certificate

Starting from Red Hat Enterprise Linux 7.2 release, the default certificate has been added to the **redhat-release** packages. This default certificate is stored in the **/etc/pki/product-default/** directory.

The Subscription Manager now searches for the list of the certificates in the **/etc/pki/product/** directory and then in the **/etc/pki/product-default/** directory. Content in the **/etc/pki/product-default/** directory is provided by **redhat-release** packages. Any certificate in the **/etc/pki/product-default/** directory that is not located in **/etc/pki/product/** is considered to be installed. The default product certificates are used until Subscription Manager fetches product certificates from the subscribed channels.

2.8. CLUSTERING AND HIGH AVAILABILITY

Read this section for a summary of changes to clustering and high availability support and relevant configuration tools between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

2.8.1. Luci replacement limitations (pcs)

In Red Hat Enterprise Linux 6, **luci** controlled both Red Hat Enterprise Linux 5 and Red Hat Enterprise Linux 6 high availability clusters.

Red Hat Enterprise Linux 7 removes **luci** and replaces it with **pcs**. **pcs** can control only Red Hat Enterprise Linux 7 pacemaker-based clusters. It cannot control Red Hat Enterprise Linux 6 rgmanager-based high availability clusters.

2.8.2. Keepalived replaces Piranha

The Load Balancer Add-On for Red Hat Enterprise Linux 7 now includes the **keepalived** service, which provides both the functionality available in **piranha** and additional functionality. **piranha** is therefore superseded by the **keepalived** service in Red Hat Enterprise Linux 7.

As a result, the configuration file and its format have changed. **keepalived** is configured in the **/etc/keepalived/keepalived.conf** file by default. Details on the configuration format and syntax expected by this file are covered in the **keepalive.conf** man page:

```
$ man keepalived.conf
```

2.8.3. Online migration limitations

Online migration from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7 is not supported for clusters.

Additionally, the Red Hat Enterprise Linux 6 high availability stack is not compatible with the Red Hat Enterprise Linux 7 high availability stack, so online migration is not supported from a Red Hat Enterprise Linux 6 to a Red Hat Enterprise Linux 7 high availability cluster.

2.8.4. New resource manager (Pacemaker)

As of Red Hat Enterprise Linux 7, **rgmanager** and **cman** are replaced by **pacemaker** and **corosync**.

Pacemaker is a high availability resource manager with many useful features.

- Detection and recovery from machine and application-level failures.
- Support for many redundancy configurations.
- Support for quorate and resource-driven clusters.
- Configurable strategies for dealing with quorum loss (when multiple machines fail).
- Support for specifying application startup and shutdown ordering, regardless of which machine the applications are on.
- Support for specifying that applications must or must not run on the same machine.
- Support for specifying that an application should be active on multiple machines.
- Support for multiple modes for applications, such as master and slave.
- Provably correct responses to any failure or cluster state.
- Responses to any situation can be tested offline, before the situation exists.

For further information about Pacemaker, see the Red Hat Enterprise Linux 7 High Availability Add-On documentation available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.8.5. New feature: resource agents

Red Hat Enterprise Linux 7 introduces resource agents that work with the Pacemaker resource manager. Resource agents abstract cluster resources and provide a standard interface for managing resources in a cluster environment. For further information about the resource agents available in Red

Hat Enterprise Linux 7, see the Red Hat Enterprise Linux 7 High Availability Add-On documentation available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

Support for IBM DB2 resource agents to drive and manage DB2 as cluster resource in High Available environments has been added in Red Hat Enterprise Linux 7.2.

2.8.6. Changed quorum implementation

qdiskd, as it was shipped in Red Hat Enterprise Linux 6, has been removed from Red Hat Enterprise Linux 7. The new quorum implementation is provided by **votequorum**, which is included in the **corosync** package, and which has been extended to replace **qdiskd** for most use cases. The extensions (**wait_for_all**, **auto_tie_breaker** and **last_man_standing**) are fully documented in the **votequorum.5** man page.

```
$ man 5 votequorum
```

2.9. DESKTOP

Read this section for a summary of changes to supported desktop user environments between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

This section covers only the major changes that users can expect from the new desktop environments in Red Hat Enterprise Linux 7. For detailed information, see the *Red Hat Enterprise Linux 7 Desktop Migration and Administration Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.9.1. New Default Desktop Environment (GNOME Classic)

GNOME Classic is the default session of the GNOME 3 desktop environment on Red Hat Enterprise Linux 7. This environment is provided as a set of extensions to the GNOME 3 desktop environment, and includes its powerful new features while retaining the familiar look and feel of GNOME 2.

In GNOME Classic, the user interface has two major components:

The top bar

> This bar across the top of the screen displays the Applications and Places menus.
> The Applications menu gives the user access to applications on the system, which are organized into a number of categories on the menu. This menu also provides access to the new **Activities Overview**, which lets you easily view your open windows, workspaces, and any messages or system notifications.
>
> The Places menu is displayed next to the Applications menu on the top bar. It gives the user quick access to important folders, for example **Downloads** or **Pictures**.

The taskbar

> The taskbar is displayed at the bottom of the screen, and features a window list, a notification icon, and a short identifier for the current workspace and the total number of available workspaces.

For a complete guide to GNOME Classic and its features, as well as the other desktop environments available in Red Hat Enterprise Linux 7, see the *Red Hat Enterprise Linux 7 Desktop Migration and Administration Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.9.2. New Desktop Environment (GNOME 3)

Red Hat Enterprise Linux 7 also supports the GNOME 3 session of the GNOME 3 desktop environment. This environment is designed for ease-of-use and user productivity. It provides great integration with online document storage services, calendars, and contact lists, so that you are always up to date.

In GNOME 3, the user interface has three major components:

The top bar

This horizontal bar at the top of the screen provides access to some basic GNOME Shell functions, such as the **Activities Overview**, clock, calendar, system status icons, and the system menu.

The Activities Overview

The **Activities Overview** lets you easily view your open windows, workspaces, and any messages or system notifications. The search bar is the easiest way to find your files, launch applications, or open configuration tools. The dash on the left-hand side shows your favourite applications, so you can access your most frequently used tools faster.

The message tray

The message tray appears as a bar across the bottom of your screen. It shows pending notifications, so you always know exactly what is happening on your system.

For a complete guide to GNOME 3 and its features, as well as the other desktop environments available in Red Hat Enterprise Linux 7, see the *Red Hat Enterprise Linux 7 Desktop Migration and Administration Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.9.3. KDE Plasma Workspaces (KDE)

Red Hat Enterprise Linux 7 provides version 4.10 of KDE Plasma Workspaces (KDE), previously known as K Desktop Environment. This updated version of KDE provides a number of enhancements, including the following:

- A polished, consistent look and feel with the default Oxygen style.
- An updated notification system (movable and closable notifications, with speed graphs) with progress visualized in the panel.
- Workspace configuration now available in **System Settings**.
- The **Activity Manager** provides the ability to add, remove, save, restore, and switch between Activities.
- Optimizations to core and user interface elements for better performance.
- Adaptive power management, with a simplified user interface and easy profile switching.
- A new **Print Manager**, which simplifies printer configuration and provides fast, accurate reporting on printer status.
- An updated **Dolphin File Manager** with navigation buttons, tabbed browsing, and improvements to metadata handling.
- An updated terminal emulator (**Konsole**) with improved tab and window control and improved interoperability.

- A new display manager, **KScreen**, which automatically can remember and restore display configuration, including resolution and relative position.
- A new applet, **Plasma Network Manager**, which makes it easy to control your network and configure network connections.

However, users should note that **Kmail** is no longer included in Red Hat Enterprise Linux 7.

2.10. DEVELOPER TOOLS

Read this section for a summary of updates to developer tool support and changes that may affect developers between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

2.10.1. Red Hat Developer Toolset

Red Hat Developer Toolset provides access to the latest stable versions of open source development tools on a separate, accelerated life cycle. It is available to Red Hat customers with an active Red Hat Developer subscription.

Red Hat Developer Toolset 2 does not currently support developing applications on Red Hat Enterprise Linux 7. However, Red Hat Developer Toolset does support developing applications on Red Hat Enterprise Linux 6, for deployment on supported minor releases of Red Hat Enterprise Linux 6 or Red Hat Enterprise Linux 7.

2.10.2. Compatibility Libraries

Red Hat Enterprise Linux 7 contains some compatibility libraries that support interfaces from previous releases of Red Hat Enterprise Linux. These libraries are included in accordance with Red Hat's Compatibility Policy, and at Red Hat's discretion. For further details, see the Red Hat Enterprise Linux 7: Application Compatibility Guide.

The following compatibility libraries are included in Red Hat Enterprise Linux 7.

Table 2.5. Compatibility libraries

Library	Last release where this interface was the default
compat-db47	Red Hat Enterprise Linux 6
compat-libcap1	Red Hat Enterprise Linux 5
compat-libf2c-34	Red Hat Enterprise Linux 4
compat-libgfortran-41	Red Hat Enterprise Linux 5
compat-openldap	Red Hat Enterprise Linux 5
libpng12	Red Hat Enterprise Linux 5
openssl098e	Red Hat Enterprise Linux 5
compat-dapl	Red Hat Enterprise Linux 5

Library	Last release where this interface was the default
compat-libtiff3	Red Hat Enterprise Linux 6
compat-libstdc++-33	Red Hat Enterprise Linux 3 (in optional repository only)

Red Hat Enterprise Linux 7 also includes the **compat-gcc-44** and **compat-gcc-44-c++** packages, which represent the system compiler shipped with Red Hat Enterprise Linux 6, and can be used along with the **compat-glibc** package for building and linking legacy software.

2.11. SECURITY AND ACCESS CONTROL

Read this section for a summary of changes to security, access control, and relevant configuration tools between Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

2.11.1. New firewall (firewalld)

In Red Hat Enterprise Linux 6, firewall capabilities were provided by the **iptables** utility, and configured either at the command line or through the graphical configuration tool, **system-config-firewall**. In Red Hat Enterprise Linux 7, firewall capabilities are still provided by **iptables**. However, administrators now interact with **iptables** through the dynamic firewall daemon, **firewalld**, and its configuration tools: **firewall-config**, **firewall-cmd**, and **firewall-applet**, which is not included in the default installation of Red Hat Enterprise Linux 7.

Because **firewalld** is dynamic, changes to its configuration can be made at any time, and are implemented immediately. No part of the firewall needs to be reloaded, so there is no unintentional disruption of existing network connections.

The primary differences between the firewall in Red Hat Enterprise Linux 6 and 7 are:

- Firewalld configuration details are not stored in **/etc/sysconfig/iptables**. Instead, configuration details are stored in various files in the **/usr/lib/firewalld** and **/etc/firewalld** directories.

- Where the firewall system in Red Hat Enterprise Linux 6 removed and re-applied all rules every time a configuration change was made, **firewalld** only applies the configuration differences. As a result, **firewalld** can change settings during runtime without losing existing connections.

For additional information and assistance configuring the firewall in Red Hat Enterprise Linux 7, see the *Red Hat Enterprise Linux 7 Security Guide* , available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.11.1.1. Migrating rules to firewalld

IMPORTANT

If you are using Red Hat Enterprise Linux 7 with another Red Hat product, such as Red Hat Enterprise Linux OpenStack Platform, it may be more appropriate to keep using **iptables** or **ip6tables** instead of moving to **firewalld**.

If you are uncertain which firewall utility to use, check your product documentation or contact Red Hat Support.

Instructions on how to disable **firewalld** and continue using **iptables** or **ip6tables** are available here: https://access.redhat.com/articles/1229233.

Red Hat Enterprise Linux 6 provided two methods of firewall configuration:

- Use the graphical **system-config-firewall** tool to configure rules. This tool stored its configuration details in the **/etc/sysconfig/system-config-firewall** file, and created configuration for the **iptables** and **ip6tables** services in the **/etc/sysconfig/iptables** and **/etc/sysconfig/ip6tables** files.
- Manually edit the **/etc/sysconfig/iptables** and **/etc/sysconfig/ip6tables** files (either from scratch, or editing an initial configuration created by **system-config-firewall**).

If you configured your Red Hat Enterprise Linux 6 firewall with **system-config-firewall**, after you upgrade your system and install **firewalld**, you can use the **firewall-offline-cmd** tool to migrate the configuration in **/etc/sysconfig/system-config-firewall** into the default zone of **firewalld**.

```
$ firewall-offline-cmd
```

However, if you manually created or edited **/etc/sysconfig/iptables** or **/etc/sysconfig/ip6tables**, after you install **firewalld**, you must either create a new configuration with **firewall-cmd** or **firewall-config**, or disable **firewalld** and continue to use the old **iptables** and **ip6tables** services. For details about creating new configurations or disabling **firewalld**, see the *Red Hat Enterprise Linux 7 Security Guide* , available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.11.2. Changes to PolicyKit

Previously, PolicyKit used key value pairs in **.pkla** files to define additional local authorizations. Red Hat Enterprise Linux 7 introduces the ability to define local authorizations with JavaScript, allowing you to script authorizations if necessary.

polkitd reads **.rules** files in lexicographic order from the **/etc/polkit-1/rules.d** and **/usr/share/polkit-1/rules.d** directories. If two files share the same name, files in **/etc** are processed before files in **/usr**. When the old **.pkla** files were processed, the last rule processed took precedence. With the new **.rules** files, the first matching rule takes precedence.

After migration, your existing rules are applied by the **/etc/polkit-1/rules.d/49-polkit-pkla-compat.rules** file. They can therefore be overridden by **.rules** files in either **/usr** or **/etc** with a name that comes before **49-polkit-pkla-compat** in lexicographic order. The simplest way to ensure that your old rules are not overridden is to begin the name of all other **.rules** files with a number greater than 49.

For further information about this, see the *Red Hat Enterprise Linux 7 Desktop Migration and Administration Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

2.11.3. Changes to user identifiers

In Red Hat Enterprise Linux 6, the base user identifier was **500**. In Red Hat Enterprise Linux 7, the base user identifier is now **1000**. This change involves replacing the **/etc/login.defs** file during the upgrade process.

If you have not modified the default **/etc/login.defs** file, the file is replaced during upgrade. The base user identifier number is changed to **1000**, and new users will be allocated user identifiers at and above 1000. User accounts created before this change retain their current user identifiers and continue to work as expected.

If you have modified the default **/etc/login.defs** file, the file is not replaced during upgrade, and the base user identifier number remains at 500.

2.11.4. Changes to libuser

As of Red Hat Enterprise Linux 7, the **libuser** library no longer supports configurations that contain both the **ldap** and **files** modules, or both the **ldap** and **shadow** modules. Combining these modules results in ambiguity in password handling, and such configurations are now rejected during the initialization process.

If you use **libuser** to manage users or groups in LDAP, you must remove the **files** and **shadow** modules from the **modules** and **create_modules** directives in your configuration file (**/etc/libuser.conf** by default).

2.11.5. Changes to opencryptoki key store

Previous versions of Red Hat Enterprise Linux used the **opencryptoki** key store version 2, which encrypted private token objects with a secure key in hardware. Red Hat Enterprise Linux 7 uses version 3, which encrypts private token objects with a clear key in software. This means that private token objects created by version 2 must be migrated before they can be used with version 3.

To migrate private token objects, perform the following procedure:

1. Update software Ensure your version of **opencryptoki** is up to date.

   ```
   # yum update -y opencryptoki
   ```

2. Verify the slot number of your token Use **pkcsconf** to determine the slot number of the token. Run the following commands as root:

   ```
   # pkcsconf -s
   # pkcsconf -t
   ```

   Note the slot number of your token. The slot description will end with **(CCA)**. The information field will identify the token as the **IBM CCA Token**.

3. Stop interface access Stop the **pkcsslotd** service and any **opencryptoki** processes.

   ```
   # systemctl stop pkcsslotd.service
   ```

   Use the following command to identify processes to stop with the **kill** utility, and then terminate the appropriate processes.

   ```
   # ps ax | grep pkcsslotd
   ```

4. Back up the data store Before you migrate, back up the CCA data store (the directory in which your tokens are stored, normally **/var/lib/opencryptoki/ccatok**). For example, make a copy of the directory.

   ```
   # cp -r /var/lib/opencryptoki/ccatok /var/lib/opencryptoki/ccatok.backup
   ```

5. Run the migration utility Change to the **/var/lib/opencryptoki/ccatok** directory and run the migration utility.

   ```
   # cd /var/lib/opencryptoki/ccatok
   # pkcscca -m v2objectsv3 -v
   ```

   When prompted, provide your Security Officer (SO) PIN and User PIN.

6. Remove outdated shared memory file Remove the **/dev/shm/var.lib.opencryptoki.ccatok** file manually, or reboot the system.

   ```
   # rm /dev/shm/var.lib.opencryptoki.ccatok
   ```

7. Go back to an operational interface access Start the **pkcsslotd** service again.

   ```
   # systemctl start pkcsslotd.service
   ```

If you encounter problems with the migration, check the following:

- Ensure you are running the commands as root, and that root is a member of the **pkcs11** group.
- Ensure that the **pkcsconf** utility is in either the **/usr/lib/pkcs11/methods/** directory or the **/usr/sbin/** directory.
- Ensure that the token data store is in the **/var/lib/opencryptoki/ccatok/** directory.
- Ensure that you have supplied a slot number and that the slot number is correct.
- Ensure that your Security Officer (SO) PIN and User PIN are correct.
- Ensure that you have write access to the current directory.

CHAPTER 3. CHANGES TO PACKAGES, FUNCTIONALITY, AND SUPPORT

Read this chapter for information about changes to the functionality or to packages provided in Red Hat Enterprise Linux 7, and changes to the support of said packages.

3.1. NEW PACKAGES

This section describes notable packages now available in Red Hat Enterprise Linux 7.

3.1.1. Preupgrade Assistant

The **Preupgrade Assistant** (**preupg**) checks for potential problems you might encounter with an upgrade from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7 before making any changes to your system. This helps you assess your chances of successfully upgrading to Red Hat Enterprise Linux 7 before the actual upgrade process begins.

The **Preupgrade Assistant** assesses the system for possible in-place upgrade limitations, such as package removals, incompatible obsoletes, name changes, deficiencies in some configuration file compatibilities, and so on. It then provides the following:

- System analysis report with proposed solutions for any detected migration issues.
- Data that could be used for "cloning" the system, if the in-place upgrade is not suitable.
- Post-upgrade scripts to finish more complex issues after the in-place upgrade.

Your system remains unchanged except for the information and logs stored by the **Preupgrade Assistant**.

For detailed instructions on how to obtain and use the **Preupgrade Assistant**, see Section 1.1.3, “Check system upgrade suitability”.

3.1.2. Red Hat Upgrade Tool

The new **Red Hat Upgrade Tool** is used after the **Preupgrade Assistant**, and handles the three phases of the upgrade process:

- **Red Hat Upgrade Tool** fetches packages and an upgrade image from a disk or server, prepares the system for the upgrade, and reboots the system.
- The rebooted system detects that upgrade packages are available and uses **systemd** and **yum** to upgrade packages on the system.
- **Red Hat Upgrade Tool** cleans up after the upgrade and reboots the system into the upgraded operating system.

Both network and disk based upgrades are supported. For detailed instructions on how to upgrade your system, see Chapter 1, *How to Upgrade*.

3.1.3. Chrony

Chrony is a new NTP client provided in the **chrony** package. It replaces the reference implementation (**ntp**) as the default NTP implementation in Red Hat Enterprise Linux 7. However, it does not support all features available in **ntp**, so **ntp** is still provided for compatibility reasons. If you require **ntp**, you must explicitly remove **chrony** and install **ntp** instead.

Chrony's timekeeping algorithms have several advantages over the **ntp** implementation.

- Faster, more accurate synchronization.
- Larger range for frequency correction.
- Better response to rapid changes in clock frequency.
- No clock stepping after initial synchronization.
- Works well with an intermittent network connection.

For more information about **chrony**, see the *Red Hat Enterprise Linux 7 System Administrators Guide* or *System Administrators Reference Guide*, available from http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

3.1.4. HAProxy

HAProxy is a TCP/HTTP reverse proxy that is well-suited to high availability environments. It requires few resources, and its event-driven architecture allows it to easily handle thousands of simultaneous connections on hundreds of instances without risking the stability of the system.

For more information about **HAProxy**, see the man page, or consult the documentation installed along with the **haproxy** package in the **/usr/share/doc/haproxy** directory.

3.1.5. Kernel-tools

The **kernel-tools** package includes a number of tools for the Linux kernel. Some tools in this package replace tools previously available in other packages. See Section 3.3, “Deprecated Packages” and Section 3.2, “Package Replacements” for details.

3.1.6. NFQUEUE (libnetfilter_queue)

Red Hat Enterprise Linux 7.1 provides the **libnetfilter_queue** package. This library enables the **NFQUEUE** iptables target, which specifies that a listening user-space application will retrieve a packet from a specified queue and determine how that packet will be handled.

3.1.7. SCAP Security Guide

The **scap-security-guide** package provides security guidance, baselines, and associated validation mechanisms for the Security Content Automation Protocol (SCAP). Previously, this package was only available through the EPEL repository (Extra Packages for Enterprise Linux). As of Red Hat Enterprise Linux 7.1, **scap-security-guide** is available in the Red Hat Enterprise Linux 7 Server (RPMS) repository.

3.1.8. Red Hat Access GUI

Red Hat Access GUI is a desktop application, which lets you find help, answers, and utilize diagnostic services using Red Hat Knowledgebase, resources, and functionality. If you have an active account on the Red Hat Customer Portal, you can access additional information and tips of the Knowledgebase easily browsable by keywords. **Red Hat Access GUI** is already installed if you select to have the GNOME Desktop installed.

For more information on the benefits, installation, and usage of this tool, see Red Hat Access GUI .

3.2. PACKAGE REPLACEMENTS

This section lists packages that have been removed from Red Hat Enterprise Linux between version 6 and version 7 alongside functionally equivalent replacement packages or alternative packages available in Red Hat Enterprise Linux 7.

Table 3.1. Replaced packages

Removed package	Replacement/Alternative	Notes
vconfig	iproute (ip tool)	Not fully compatible.
module-init-tools	kmod	
openoffice.org	libreoffice	
man	man-db	
ext2 and ext3 filesystem driver	ext4 filesystem driver	
openais	corosync	Functionality wrapped by the Red Hat Enterprise Linux HA stack.
jwhois	whois	Output format differs.
libjpeg	libjpeg-turbo	
gpxe	ipxe	Fork of **gpxe**.
cpuspeed	kernel, kernel-tools (cpupower, cpupower.service)	Now configured in **/etc/sysconfig/cpupower**. No longer includes user-space scaling daemon; use kernel governors if necessary.
nc	nmap-ncat	
procps	procps-ng	
openswan	libreswan	
arptables_jf	arptables	
gcj	OpenJDK	Do not compile Java apps to native code with **gcj**.

Removed package	Replacement/Alternative	Notes
32-bit x86 as installation architecture	AMD64 and Intel 64	Applications will still run with compatibility libraries. Test your applications on 64-bit Red Hat Enterprise Linux 6. If 32-bit x86 boot support is required, continue to use Red Hat Enterprise Linux 6.
Power 6 PPC support		Continue to use Red Hat Enterprise Linux 5 or Red Hat Enterprise Linux 6
Matahari	CIM-based management	
ecryptfs	Use existing LUKS/dm-crypt block-based encryption	Migration is not available for encrypted file systems; encrypted data must be recreated.
evolution-exchange	evolution-mapi/evolution-ews	
TurboGears2 web application stack		
openmotif22	motif	Rebuild applications against the current Motif version.
webalizer web anayltics tool		Other web analytics tools are superior.
compiz window manager	gnome-shell	
Eclipse developer toolset		Eclipse is now offered in the Developer Toolset offering.
Qpid and QMF		Qpid and QMF are available in the MRG offering.
amtu		Common Criteria certifications no longer require this tool.
pidgin frontends	empathy	
perl-suidperl	perl	This functionality has been removed in upstream perl.

Removed package	Replacement/Alternative	Notes
pam_passwdqc, pam_cracklib	libpwquality, pam_pwquality	Not fully compatible.
HAL library and daemon	udev	
ConsoleKit library and daemon	systemd	Not fully compatible.
system-config-network	nm-connection-editor, nmcli	
thunderbird	evolution	
system-config-firewall	firewalld	
busybox	normal utilities	
KVM/virt packages (in ComputeNode)	KVM/virt equipped variant such as a Server variant	
abyssinica-fonts	sil-abyssinica-fonts	
axis	java-1.7.0-openjdk	
ccs	pcs	Not fully compatible.
cjkuni-fonts-common	cjkuni-uming-fonts	
classpath-jaf	java-1.7.0-openjdk	
classpath-mail	javamail	Not fully compatible.
cman	corosync	
control-center-extra	control-center	
db4-cxx	libdb4-cxx	
db4-devel	libdb4-devel	
db4-utils	libdb4-utils	
desktop-effects	control-center	
DeviceKit-power	upower	Not fully compatible.
dracut-kernel	dracut	

Removed package	Replacement/Alternative	Notes
eggdbus	glib2	Not fully compatible.
fcoe-target-utils	targetcli	See Section 2.6.3, “Target Management with targetcli” for details.
febootstrap	supermin	
gcc-java	java-1.7.0-openjdk-devel	
GConf2-gtk	GConf2	
gdm-plugin-fingerprint	gdm	
gdm-plugin-smartcard	gdm	
gdm-user-switch-applet	gnome-shell	Not fully compatible.
geronimo-specs	geronimo-parent-poms	
geronimo-specs-compat	geronimo-jms, geronimo-jta	Not fully compatible.
gimp-help-browser	gimp	Not fully compatible.
gnome-applets	gnome-classic-session	Not fully compatible.
gnome-keyring-devel	gnome-keyring	
gnome-mag	gnome-shell	Not fully compatible.
gnome-python2-applet	pygtk2	Not fully compatible.
gnome-speech	speech-dispatcher	Not fully compatible.
gpxe-roms-qemu	ipxe-roms-qemu	
hal	systemd	Not fully compatible.
hal-devel	systemd-devel	Not fully compatible.
ibus-gtk	ibus-gtk2	
ibus-table-cangjie	ibus-table-chinese-cangjie	
ibus-table-erbi	ibus-table-chinese-erbi	

Removed package	Replacement/Alternative	Notes
ibus-table-wubi	ibus-table-chinese-wubi-haifeng	
jakarta-commons-net	apache-commons-net	
java-1.5.0-gcj	java-1.7.0-openjdk, java-1.7.0-openjdk-headless	Not fully compatible.
java-1.5.0-gcj-devel	java-1.7.0-openjdk-devel	Not fully compatible.
java-1.5.0-gcj-javadoc	java-1.7.0-openjdk-javadoc	Not fully compatible.
junit4	junit	
jwhois	whois	
kabi-whitelists	kernel-abi-whitelists	
kdeaccessibility-libs	kdeaccessibility	
kdebase-devel	kde-baseapps-devel	
kdebase-workspace-wallpapers	kde-wallpapers	
kdelibs-experimental	kdelibs	
kdesdk-libs	kate-libs, kdesdk-kmtrace-libs, kdesdk-kompare	Not fully compatible.
kdesdk-utils	kdesdk-poxml	
krb5-auth-dialog	gnome-online-accounts	Not fully compatible.
lldpad-libs	lldpad	
lslk	util-linux	Not fully compatible.
luci	pcs	See Section 2.8, “Clustering and High Availability” for details.
man-pages-uk	man-pages	
mingetty	util-linux	Not fully compatible.
modcluster	pcs	Not fully compatible.

Removed package	Replacement/Alternative	Notes
mod_perl	mod_fcgid	Not compatible with httpd 2.4.
m17n-contrib-*	m17n-contrib	
m17n-db-*	m17n-db, m17n-db-extras	
NetworkManager-gnome	nm-connection-editor, network-manager, applet	
nss_db	glibc	Not fully compatible.
openais	corosync	
openaislib	corosynclib	
openaislib-devel	corosynclib-devel	
PackageKit-gtk-module	PackageKit-gtk3-module	Not fully compatible.
polkit-desktop-policy	polkit	
pulseaudio-libs-zeroconf	pulseaudio-libs	Not fully compatible.
qt-sqlite	qt	
rdesktop	xfreerdp	
Red_Hat_Enterprise_Linux-Release_Notes-6-*	Red_Hat_Enterprise_Linux-Release_Notes-7-*	
redhat-lsb-compat	redhat-lsb-core	
rgmanager	pacemaker	See Section 2.8, “Clustering and High Availability” for details.
rhythmbox-upnp	rhythmbox	
ricci	pcs	See Section 2.8, “Clustering and High Availability” for details.
samba4*	samba*	See Section 2.7.6.3, “Samba” for details.
sbm-cim-client	sbm-cim-client2	Not fully compatible.

Removed package	Replacement/Alternative	Notes
scsi-target-utils	targetcli	See Section 2.6.3, “Target Management with targetcli” for details.
seekwatcher	iowatcher	
spice-client	virt-viewer	Not fully compatible.
system-config-lvm	gnome-disk-utility	Not fully compatible.
texlive-*	texlive	
tex-cm-lgc	texlive-cm-lgc	
tex-kerkis	texlive-kerkis	
texlive-texmf-dvips	texlive-dvips	
texlive-texmf-latex	texlive-latex	
tomcat6	tomcat	
tomcat6-el-2.1-api	tomcat-el-2.2-api	
tomcat6-jsp-2.1-api	tomcat-jsp-2.2-api	
tomcat6-lib	tomcat-lib	
totem-upnp	totem	
udisks	udisks2	Not fully compatible.
un-core-batang-fonts	nhn-nanum-myeongjo-fonts	
un-core-dinaru-fonts, un-core-graphic-fonts	nhn-nanum-gothic-fonts	Not fully compatible.
un-core-dotum-fonts	nhn-nanum-gothic-fonts	
un-core-fonts-common	nhn-nanum-fonts-common	Not fully compatible.
un-core-gungseo-fonts	nhn-nanum-brush-fonts	Not fully compatible.
un-core-pilgi-fonts	nhn-nanum-pen-fonts	Not fully compatible.

Removed package	Replacement/Alternative	Notes
unique	unique3, glib2	Not fully compatible.
unique-devel	unique3-devel	Not fully compatible.
unix2dos	dos2unix	
vgabios	seavgabios-bin	
w3m	text-www-browser	Not fully compatible.
xmlrpc3-*	xmlrpc-*	
xorg-x11-drv-apm	xorg-x11-drv-fbdev, xorg-x11-drv-vesa	
xorg-x11-drv-ast, xorg-x11-drv-cirrus, xorg-x11-drv-mga	xorg-x11-drv-modesetting	
xorg-x11-drv-ati-firmware	linux-firmware	
xorg-x11-drv-elographics, xorg-x11-drv-glint, xorg-x11-drv-i128, xorg-x11-drv-i740, xorg-x11-drv-mach64, xorg-x11-drv-rendition, xorg-x11-drv-r128, xorg-x11-drv-savage, xorg-x11-drv-siliconmotion, xorg-x11-drv-sis, xorg-x11-drv-sisusb, xorg-x11-drv-s3virge, xorg-x11-drv-tdfx, xorg-x11-drv-trident, xorg-x11-drv-voodoo, xorg-x11-drv-xgi	xorg-x11-drv-fbdev, xorg-x11-drv-vesa	
xorg-x11-drv-nv	xorg-x11-drv-nouveau	
xorg-x11-twm	metacity	Not fully compatible.
xorg-x11-xdm	gdm	Not fully compatible.
yum-plugin-downloadonly	yum	

3.3. DEPRECATED PACKAGES

The packages listed in this section are considered deprecated as of Red Hat Enterprise Linux 7. These packages still work, and remain supported, but Red Hat no longer recommends their use.

Table 3.2. Package deprecations

Functionality/Package	Alternative	Migration Notes
ext2 file system support	ext3, ext4	ext4 can be used for ext2 and ext3 file systems.
sblim-sfcb	tog-pegasus	
Legacy RHN Hosted registration	subscription-manager and Subscription Asset Manager	
acpid	systemd	
evolution-mapi	evolution-ews	Please migrate from Microsoft Exchange Server 2003 machines
gtkhtml3	webkitgtk3	
sendmail	postfix	
edac-utils and mcelog	rasdaemon	
libcgroup	systemd	cgutils will continue to exist in Red Hat Enterprise Linux 7.0 but systemd is evolving capabilities to enable customers to migrate in later releases
lvm1	lvm2	
lvm2mirror and cmirror	lvm2 raid1	

3.4. REMOVED PACKAGES

The following packages have been removed from Red Hat Enterprise Linux between version 6 and version 7 and are no longer supported. Some of these packages may have functionally equivalent replacements available; see Section 3.2, “Package Replacements” for details.

- amtu
- ant-antlr
- ant-apache-bcel
- ant-apache-bsf
- ant-apache-log4j
- ant-apache-oro
- ant-apache-regexp

- ant-apache-resolver
- ant-commons-logging
- ant-commons-net
- ant-javamail
- ant-jdepend
- ant-jsch
- ant-junit
- ant-nodeps
- ant-swing
- ant-trax
- apache-jasper
- apache-tomcat-apis
- apr-util-ldap
- arts
- arts-devel
- aspell
- atmel-firmware
- at-spi
- at-spi-python
- audiofile
- audit-viewer
- avahi-tools
- avahi-ui
- avalon-framework
- avalon-logkit
- batik
- brasero
- brasero-libs
- brasero-nautilus

- bsf
- busybox
- b43-fwcutter
- b43-openfwwf
- cas
- cdparanoia
- cdrdao
- cjet
- cloog-ppl
- cluster-cim
- cluster-glue
- cluster-glue-libs
- cluster-glue-libs-devel
- clusterlib
- clusterlib-devel
- cluster-snmp
- cman
- compat-db42
- compat-db43
- compat-libstdc++-296
- compat-libtermcap
- compat-openmpi
- compat-openmpi-psm
- compat-opensm-libs
- compiz
- compiz-gnome
- coreutils-libs
- cracklib-python
- cronie-noanacron

- ctan-cm-lgc-fonts-common
- ctan-cm-lgc-roman-fonts
- ctan-cm-lgc-sans-fonts
- ctan-cm-lgc-typewriter-fonts
- ctan-kerkis-fonts-common
- ctan-kerkis-sans-fonts
- ctan-kerkis-serif-fonts
- ctapi-common
- cvs-inetd
- c2050
- c2070
- dash
- dbus-c+
- dbus-qt
- devhelp
- dmz-cursor-themes
- dtach
- dvd+rw-tools
- eclipse-birt
- eclipse-callgraph
- eclipse-cdt
- eclipse-dtp
- eclipse-emf
- eclipse-gef
- eclipse-changelog
- eclipse-jdt
- eclipse-linuxprofilingframework
- eclipse-mylyn
- eclipse-mylyn-cdt

- eclipse-mylyn-java
- eclipse-mylyn-pde
- eclipse-mylyn-trac
- eclipse-mylyn-webtasks
- eclipse-mylyn-wikitext
- eclipse-nls
- eclipse-nls-ar
- eclipse-nls-bg
- eclipse-nls-ca
- eclipse-nls-cs
- eclipse-nls-da
- eclipse-nls-de
- eclipse-nls-el
- eclipse-nls-es
- eclipse-nls-et
- eclipse-nls-fa
- eclipse-nls-fi
- eclipse-nls-fr
- eclipse-nls-he
- eclipse-nls-hi
- eclipse-nls-hu
- eclipse-nls-id
- eclipse-nls-it
- eclipse-nls-ja
- eclipse-nls-ko
- eclipse-nls-ku
- eclipse-nls-mn
- eclipse-nls-nl
- eclipse-nls-no

- eclipse-nls-pl
- eclipse-nls-pt
- eclipse-nls-pt_BR
- eclipse-nls-ro
- eclipse-nls-ru
- eclipse-nls-sk
- eclipse-nls-sl
- eclipse-nls-sq
- eclipse-nls-sr
- eclipse-nls-sv
- eclipse-nls-tr
- eclipse-nls-uk
- eclipse-nls-zh
- eclipse-nls-zh_TW
- eclipse-oprofile
- eclipse-pde
- eclipse-platform
- eclipse-rcp
- eclipse-rpm-editor
- eclipse-rse
- eclipse-subclipse
- eclipse-subclipse-graph
- eclipse-svnkit
- eclipse-swt
- eclipse-valgrind
- ecryptfs-utils
- evolution-data-server-doc
- fakechroot
- fakechroot-libs

- fence-virt
- fence-virtd-checkpoint
- file-devel
- firstaidkit
- firstaidkit-engine
- firstaidkit-gui
- foghorn
- fop
- gamin-devel
- gamin-python
- gconfmm26
- ggz-base-libs
- glade3
- gnome-disk-utility-libs
- gnome-disk-utility-ui-libs
- gnome-doc-utils
- gnome-doc-utils-stylesheets
- gnome-games
- gnome-media
- gnome-media-libs
- gnome-pilot
- gnome-pilot-conduits
- gnome-power-manager
- gnome-python2-bugbuddy
- gnome-python2-extras
- gnome-python2-gtkhtml2
- gnome-python2-libegg
- gnome-python2-libwnck
- gnome-python2-rsvg

- gnome-themes
- gnome-user-share
- gnome-vfs2-devel
- gnome-vfs2-smb
- graphviz-perl
- groff
- gsl-static
- gstreamer-python
- gthumb
- gtk+extra
- gtkhtml2
- gtksourceview2
- gtk2-engines
- guile
- gvfs-afc
- gvfs-archive
- hal-info
- hal-libs
- hal-storage-addon
- htdig
- hypervkvpd
- ibus-table-additional
- icedax
- icu4j-eclipse
- ipa-pki-ca-theme
- ipa-pki-common-theme
- ipw2100-firmware
- ipw2200-firmware
- jakarta-commons-discovery

- jakarta-commons-el
- jasper
- java_cup
- jdepend
- jetty-eclipse
- jsch
- jzlib
- kabi-yum-plugins
- kcoloredit
- kcoloredit-doc
- kdeadmin
- kdeartwork-screensavers
- kdebase-workspace-akonadi
- kdebase-workspace-python-applet
- kdegames
- kdegraphics
- kde-i18n-Arabic
- kde-i18n-Bengali
- kde-i18n-Brazil
- kde-i18n-British
- kde-i18n-Bulgarian
- kde-i18n-Catalan
- kde-i18n-Czech
- kde-i18n-Danish
- kde-i18n-Dutch
- kde-i18n-Estonian
- kde-i18n-Finnish
- kde-i18n-French
- kde-i18n-German

- kde-i18n-Greek
- kde-i18n-Hebrew
- kde-i18n-Hindi
- kde-i18n-Hungarian
- kde-i18n-Chinese
- kde-i18n-Chinese-Big5
- kde-i18n-Icelandic
- kde-i18n-Italian
- kde-i18n-Japanese
- kde-i18n-Korean
- kde-i18n-Lithuanian
- kde-i18n-Norwegian
- kde-i18n-Norwegian-Nynorsk
- kde-i18n-Polish
- kde-i18n-Portuguese
- kde-i18n-Punjabi
- kde-i18n-Romanian
- kde-i18n-Russian
- kde-i18n-Serbian
- kde-i18n-Slovak
- kde-i18n-Slovenian
- kde-i18n-Spanish
- kde-i18n-Swedish
- kde-i18n-Tamil
- kde-i18n-Turkish
- kde-i18n-Ukrainian
- kdelibs-apidocs
- kdelibs3
- kdelibs3-devel

- kde-l10n-Bengali-India
- kde-l10n-Frisian
- kde-l10n-Gujarati
- kde-l10n-Chhattisgarhi
- kde-l10n-Kannada
- kde-l10n-Kashubian
- kde-l10n-Kurdish
- kde-l10n-Macedonian
- kde-l10n-Maithili
- kde-l10n-Malayalam
- kde-l10n-Marathi
- kdemultimedia
- kdemultimedia-devel
- kdemultimedia-libs
- kdenetwork
- kdesdk
- kdesdk-libs
- kdeutils
- kdewebdev
- kdewebdev-libs
- kernel-debug
- kernel-debug-devel
- kernel-doc
- kiconedit
- kipi-plugins
- kipi-plugins-libs
- kmid
- kmid-common
- konq-plugins-doc

- krb5-appl
- kross-python
- ksig
- ksig-doc
- k3b
- k3b-common
- k3b-libs
- libao-devel
- libart_lgpl-devel
- libbonobo-devel
- libbonoboui-devel
- libburn
- libcroco-devel
- libdc1394
- libdiscid
- libesmtp-devel
- libexif-devel
- libgail-gnome
- libgcj
- libgcj-devel
- libgcj-src
- libglademm24
- libglade2-devel
- libgnomecanvas-devel
- libgnome-devel
- libgnomeui-devel
- libgphoto2-devel
- libgpod
- libgsf-devel

- libgxim
- libIDL-devel
- libidn-devel
- libisofs
- libitm
- libldb-devel
- libmatchbox
- libmtp
- libmusicbrainz
- libmusicbrainz3
- libnih
- liboil
- libopenraw-gnome
- libpanelappletmm
- libproxy-bin
- libproxy-python
- libreport-compat
- libreport-plugin-mailx
- libreport-plugin-reportuploader
- librtas (32-bit only)
- libselinux-ruby
- libservicelog (32-bit only)
- libsexy
- libtalloc-devel
- libtdb-devel
- libtevent-devel
- libtidy
- libvpd (32-bit only)
- libwnck

- libXdmcp-devel
- log4cpp
- lpg-java-compat
- lucene
- lucene-contrib
- lx
- lynx
- MAKEDEV
- matchbox-window-manager
- mcstrans
- mesa-dri1-drivers
- min12xxw
- mod_auth_mysql
- mod_auth_pgsql
- mod_authz_ldap
- mod_dnssd
- mrtg-libs
- mvapich-psm-static
- mx4j
- nspluginwrapper
- openct
- openhpi-subagent
- openssh-askpass
- ORBit2-devel
- osutil
- oxygen-cursor-themes
- PackageKit-yum-plugin
- paktype-fonts-common
- pam_passwdqc

- pbm2l2030
- pbm2l7k
- pcmciautils
- pcsc-lite-openct
- perl-BSD-Resource
- perl-Cache-Memcached
- perl-Class-MethodMaker
- perl-Config-General
- perl-Crypt-PasswdMD5
- perl-Frontier-RPC
- perl-Frontier-RPC-doc
- perl-Perlilog
- perl-String-CRC32
- perl-suidperl
- perl-Text-Iconv
- perl-Time-HiRes
- perl-YAML-Syck
- pessulus
- pilot-link
- pinentry-gtk
- piranha
- pki-symkey
- plpa-libs
- plymouth-gdm-hooks
- plymouth-theme-rings
- plymouth-utils
- policycoreutils-newrole
- policycoreutils-sandbox
- ppl

- prelink
- printer-filters
- psutils
- ptouch-driver
- pulseaudio-module-gconf
- pycairo-devel
- pygobject2-codegen
- pygobject2-devel
- pygobject2-doc
- pygtksourceview
- pygtk2-codegen
- pygtk2-devel
- pygtk2-doc
- pychart
- PyOpenGL [1]
- python-beaker
- python-Coherence
- python-crypto
- python-decoratortools
- python-enchant
- python-formencode
- python-fpconst
- python-genshi
- python-gtkextra
- python-cheetah
- python-ipaddr
- python-iwlib
- python-libguestfs [2]
- python-louie

- python-mako
- python-markdown
- python-markupsafe
- python-matplotlib
- python-myghty
- python-paramiko
- python-paste
- python-paste-deploy
- python-paste-script
- python-peak-rules
- python-peak-util-addons
- python-peak-util-assembler
- python-peak-util-extremes
- python-peak-util-symbols
- python-prioritized-methods
- python-pygments
- python-pylons
- python-qpid
- python-qpid-qmf
- python-repoze-tm2
- python-repoze-what
- python-repoze-what-plugins-sql
- python-repoze-what-pylons
- python-repoze-what-quickstart
- python-repoze-who
- python-repoze-who-friendlyform
- python-repoze-who-plugins-sa
- python-repoze-who-testutil
- python-routes

- python-saslwrapper
- python-sexy
- python-sqlalchemy
- python-tempita
- python-toscawidgets
- python-transaction
- python-turbojson
- python-tw-forms
- python-twisted
- python-twisted-conch
- python-twisted-core
- python-twisted-lore
- python-twisted-mail
- python-twisted-names
- python-twisted-news
- python-twisted-runner
- python-twisted-web
- python-twisted-words
- python-weberror
- python-webflash
- python-webhelpers
- python-webob
- python-webtest
- python-zope-filesystem
- python-zope-interface
- python-zope-sqlalchemy
- pywebkitgtk
- pyxf86config
- qpid-cpp-client

- qpid-cpp-client-ssl
- qpid-cpp-server
- qpid-cpp-server-ssl
- qpid-qmf
- qpid-tests
- qpid-tools
- qt-doc
- raptor
- rgmanager
- rome
- ruby-devel
- ruby-qpid
- ruby-qpid-qmf
- sabayon
- sabayon-apply
- sac
- samba-winbind-clients
- samba4
- samba4-client
- samba4-common
- samba4-dc
- samba4-dc-libs
- samba4-devel
- samba4-pidl
- samba4-swat
- samba4-test
- samba4-winbind
- samba4-winbind-clients
- samba4-winbind-krb5-locator

- saslwrapper
- sat4j
- saxon
- sblim-cmpi-dhcp
- sblim-cmpi-dns
- sblim-cmpi-samba
- sblim-tools-libra
- scenery-backgrounds
- seabios
- selinux-policy-minimum
- selinux-policy-mls
- setools-console
- sgabios-bin
- sigar
- sinjdoc
- smp_utils
- SOAPpy
- sound-juicer
- strigi-devel
- subscription-manager-migration-data
- subversion-javahl
- svnkit
- system-config-firewall
- system-config-firewall-tui
- system-config-network-tui
- system-config-services
- system-config-services-docs
- system-gnome-theme
- system-icon-theme

- taskjuggler
- tbird
- terminus-fonts
- tidy
- tigervnc-server
- tix
- tkinter
- trilead-ssh2
- tsclient
- tunctl
- TurboGears2
- unicap
- vorbis-tools
- wacomexpresskeys
- wdaemon
- webalizer
- webkitgtk
- ws-commons-util
- wsdl4j
- xfig-plain
- xfsprogs-devel
- xfsprogs-qa-devel
- xguest
- xmldb-api
- xmldb-api-sdk
- xmlgraphics-commons
- xorg-x11-apps
- xorg-x11-drv-acecad
- xorg-x11-drv-aiptek

- xorg-x11-drv-fpit
- xorg-x11-drv-hyperpen
- xorg-x11-drv-keyboard
- xorg-x11-drv-mouse
- xorg-x11-drv-mutouch
- xorg-x11-drv-openchrome
- xorg-x11-drv-penmount
- xorg-x11-server-Xephyr
- xsane
- xz-lzma-compat
- zd1211-firmware

3.5. REMOVED DRIVERS

The following drivers have been removed from Red Hat Enterprise Linux between version 6 and version 7 and are no longer supported.

- 3c574_cs.ko
- 3c589_cs.ko
- 3c59x.ko
- 8390.ko
- acenic.ko
- amd8111e.ko
- avma1_cs-ko [3]
- avm_cs.ko
- axnet_cs.ko
- b1pcmpcia.ko
- bluecard_cs-ko
- bt3c_cs.ko
- btuart_cs.ko
- can-dev.ko
- cassini.ko

- cdc-phonet.ko
- cm4000_cs.ko
- cm4040_cs.ko
- cxgb.ko
- de2104x.ko
- de4x5.ko
- dl2k.ko
- dmfe.ko
- dtl1_cs.ko
- e100.ko
- elsa_cs.ko
- ems_pci.ko
- ems_usb.ko
- fealnx.ko
- fmvj18x_cs.ko
- forcedeth.ko
- ipwireless.ko
- ixgb.ko
- kvaser_pci.ko
- myri10ge.ko
- natsemi.ko
- ne2k-pci.ko
- niu.ko
- nmclan_cs.ko
- ns83820.ko
- parport_cs.ko
- pata_pcmcia.ko
- pcnet_cs.ko
- pcnet32.ko

- pppol2tp.ko
- r6040.ko
- s2io.ko
- sc92031.ko
- sdricoh_cs.ko
- sedlbauer_cs.ko
- serial_cs.ko
- sis190.ko
- sis900.ko
- sja1000_platform.ko
- sja1000.ko
- smc91c92_cs.ko
- starfire.ko
- sundance.ko
- sungem_phy.ko
- sungem.ko
- sunhme.ko
- tehuti.ko
- teles_cs.ko
- tlan.ko
- tulip.ko
- typhoon.ko
- uli526x.ko
- vcan.ko
- via-rhine.ko
- via-velocity.ko
- vxge.ko
- winbond-840.ko
- xirc2ps_cs.ko

- xircom_cb.ko

3.6. DEPRECATED DRIVERS

For information about deprecated drivers in Red Hat Enterprise Linux 7, see the most recent version of Release Notes on the Red Hat Customer Portal .

[1] Removed in Red Hat Enterprise Linux 7.0, replaced in Red Hat Enterprise Linux 7.1. Added to Optional channel in Red Hat Enterprise Linux 7.3. For more information about Optinal channel, see this solution article.

[2] Moved to the Optional repository for Red Hat Enterprise Linux 7.0, back in the base channel since Red Hat Enterprise Linux 7.1.

[3] The PCMCIA is not supported in Red Hat Enterprise Linux 7. It has been superseded by new technologies, including USB.

CHAPTER 4. CUSTOMER PORTAL LABS RELEVANT FOR MIGRATION

Red Hat Customer Portal Labs are tools designed to help you improve performance, troubleshoot issues, identify security problems, and optimize configuration. This appendix provides an overview of Red Hat Customer Portal Labs relevant to migration. All Red Hat Customer Portal Labs are available at http://access.redhat.com/labs/.

Red Hat Enterprise Linux Upgrade Helper

The Red Hat Enterprise Linux Update Helper is a tool that helps you upgrade your Red Hat Enterprise Linux from version 6.5/6.6/6.7/6.8/6.9 to version 7.x. The only information that you need to provide is your upgrade path. This application shows you:

- the basic steps to upgrade Red Hat Enterprise Linux
- extra steps that prevent known issues specific to your upgrade scenario

This application supports the following upgrade paths:

- 6.5 to 7.4
- 6.6 to 7.4
- 6.7 to 7.4
- 6.8 to 7.4
- 6.9 to 7.4

Product Life Cycle Checker

The Product Life Cycle Checker is a tool for viewing Red Hat products' life-cycle information, including General Availability, End of Support, and End of Life. With this tool, it is possible to choose multiple products and view their dates.