



Red Hat Enterprise Linux for Real Time 7 Installation Guide

Installation information for Red Hat Enterprise Linux for Real Time

Radek Bíba
Lana Brindley

David Ryan
Alison Young

Cheryn Tan

Red Hat Enterprise Linux for Real Time 7 Installation Guide

Installation information for Red Hat Enterprise Linux for Real Time

Radek Bíba
Red Hat Customer Content Services
rbiba@redhat.com

David Ryan
Red Hat Customer Content Services
dryan@redhat.com

Cheryn Tan
Red Hat Customer Content Services

Lana Brindley
Red Hat Customer Content Services

Alison Young
Red Hat Customer Content Services

Legal Notice

Copyright © 2015 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This book describes how to install Red Hat Enterprise Linux for Real Time.

Table of Contents

Preface	2
Chapter 1. Why Use Red Hat Enterprise Linux for Real Time to Optimize Latency?	3
Chapter 2. Download	5
2.1. Installing Red Hat Enterprise Linux for Real Time Using Yum	5
2.2. Installing Red Hat Enterprise Linux for Real Time Using Diskless Boot	6
2.3. Available RPM Packages	8
2.4. Post-Installation Instructions	10
Chapter 3. Red Hat Enterprise Linux for Real Time Kernel Tuning	12
Chapter 4. More Information	14
4.1. Reporting Bugs	14
Revision History	15

Preface

This book contains basic installation and tuning information about Red Hat Enterprise Linux for Real Time.

Many industries and organizations need extremely high performance computing and may require low and predictable latency, especially in the financial and telecommunications industries. Latency, or response time, is defined as the time between an event and system response and is generally measured in microseconds (μs).

For most applications running under a Linux environment, basic performance tuning can improve latency sufficiently. For those industries where latency not only needs to be low, but also accountable and predictable, Red Hat has now developed a 'drop-in' kernel replacement that provides this. Red Hat Enterprise Linux for Real Time is distributed as part of Red Hat Enterprise Linux 7 and provides seamless integration with Red Hat Enterprise Linux 7. Red Hat Enterprise Linux for Real Time offers clients the opportunity to measure, configure and record latency times within their organization.

Chapter 1. Why Use Red Hat Enterprise Linux for Real Time to Optimize Latency?

For anyone considering evaluating the performance benefits of the Red Hat Enterprise Linux for Real Time kernel, it is crucial to understand both the importance of tuning and how to establish the right performance expectations.

Red Hat Enterprise Linux for Real Time is designed to be used on well-tuned systems for applications with extremely high determinism requirements. Kernel system tuning offers the vast majority of the improvement in determinism. For example, in many workloads thorough system tuning improves consistency of results by around 90%. This is why we typically recommend that customers first perform the suggested system tuning of standard Red Hat Enterprise Linux to see if it meets their objectives, before using Red Hat Enterprise Linux for Real Time.

System tuning is just as important when using the realtime kernel as it is for the standard kernel. In fact, if you simply took an untuned system running the standard kernel and substitute the realtime kernel for the stock kernel supplied as part of the Red Hat Enterprise Linux 7 release, you are unlikely to notice any benefit. Standard tuning will yield 90% of the latency gains. The realtime kernel provides the last 10% of latency reduction required by the most demanding workloads.

Establishing the right performance expectations refers to the fact that the realtime kernel is not a panacea. Its objective is consistent, low-latency determinism offering predictable response times. There is some additional kernel overhead associated with the realtime kernel. This is due primarily to handling hardware interrupts in separately scheduled threads. The increased overhead in some workloads results in some degradation in overall throughput. The exact amount is very workload dependent, ranging from 0% to 30%. However, it is the cost of determinism.

For typical workloads with kernel latency requirements in the millisecond (ms) range, then the standard Red Hat Enterprise Linux 7 kernel is sufficient. If your workload has stringent low-latency determinism requirements for core kernel features such as interrupt handling and process scheduling in the microsecond (μ s) range, then the realtime kernel is for you.

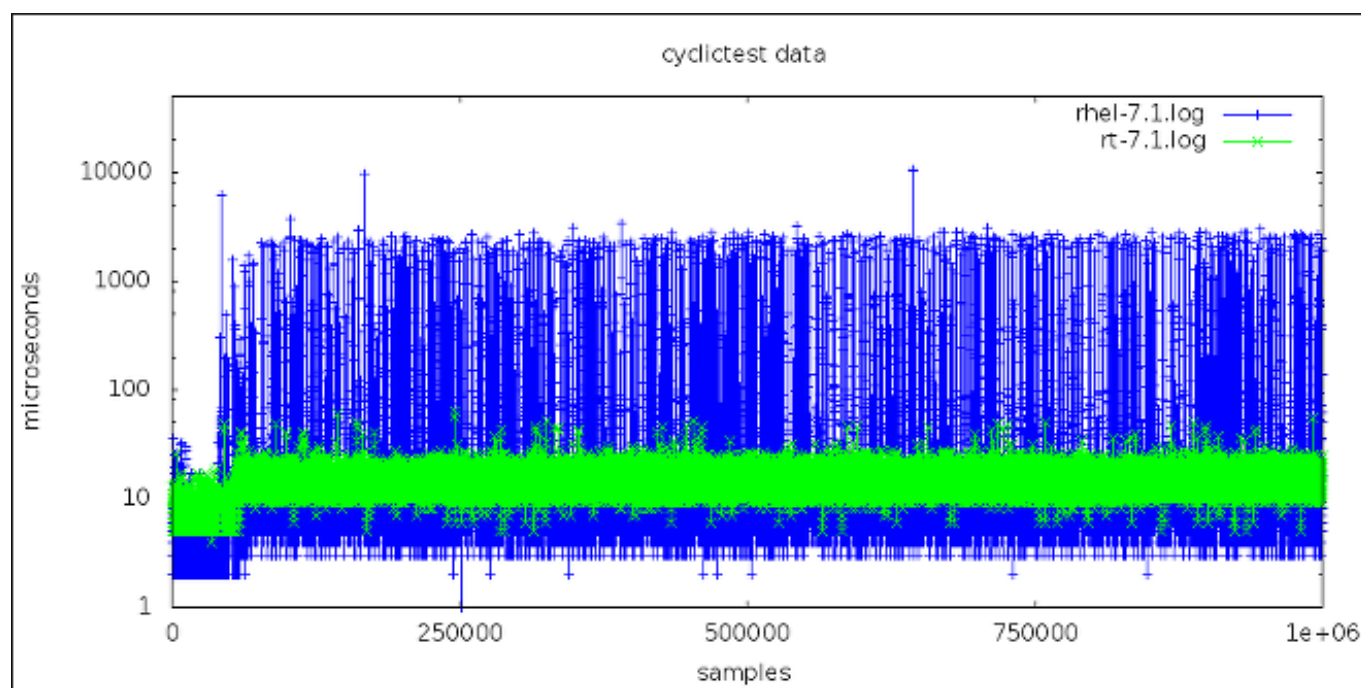


Figure 1.1. Benefit of Using Realtime over Standard Kernel System Tuning.

This graph compares a million samples of machines using the Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux for Real Time kernels respectively. The blue points in this graph represent the system response time (in microseconds) of machines running a tuned Red Hat Enterprise Linux 7 kernel. The green points in the graph represent the system response time of machines running a tuned realtime kernel. It is clear from this graph that the response time of the realtime kernel is very consistent, in contrast to the standard kernel, which has greater variability with points scattered across the graph.

Chapter 2. Download

System Requirements

The only prerequisite of installing Red Hat Enterprise Linux for Real Time is that you have installed Red Hat Enterprise Linux 7.1 on an AMD64 or Intel 64 system.

Installation Options

The following sections describe how to install Red Hat Enterprise Linux for Real Time. Alternatively, an ISO image containing Red Hat Enterprise Linux for Real Time is available for download from the [Customer Portal](#). You can use this ISO image to obtain all the RPM packages that Red Hat Enterprise Linux for Real Time consists of; however, because this is not a bootable ISO image, you cannot use it to create a bootable USB or CD medium.

2.1. Installing Red Hat Enterprise Linux for Real Time Using Yum

Make sure you have [registered your system](#). Then follow these instructions:

1. Enable the Red Hat Enterprise Linux for Real Time repository:

```
~]# subscription-manager repos --enable rhel-7-server-rt-rpms
```

2. Install the Red Hat Enterprise Linux for Real Time package group:

```
~]# yum groupinstall RT
```

This group installs several packages:

- ✦ *kernel-rt* is the Red Hat Enterprise Linux for Real Time kernel package.
- ✦ *rt-setup* sets up the basic environment required by Red Hat Enterprise Linux for Real Time.
- ✦ *rt-tests* contains programs that test various RT features.
- ✦ *rtcheck* is a program that tests the running system for Red Hat Enterprise Linux for Real Time capabilities.
- ✦ *rtctl* is a startup script that sets the priorities of the various kernel threads.
- ✦ *rteval* evaluates system suitability for Red Hat Enterprise Linux for Real Time.
- ✦ *rteval-common* provides common files for *rteval*.
- ✦ *rteval-loads* provides source code for *rteval* loads.

Additionally, *tuna* is a package containing a tool used to help tune the Red Hat Enterprise Linux for Real Time workload, greatly automating CPU isolation and thread affinity operations from the command line or the GUI. This package is available in the base Red Hat Enterprise Linux 7 repository. For information on using **tuna**, see [Chapter 3, Red Hat Enterprise Linux for Real Time Kernel Tuning](#).

You can check the installation location and that the components have been installed successfully by using the **rpm -q1** command:

```
~]# rpm -q1 rt-setup
```

```

/etc/security/limits.d/realtime.conf
/etc/sysconfig/rt-setup
/etc/systemd/system/rt-setup.service
/etc/udev/rules.d/99-rhel-rt.rules
/usr/bin/rt-setup
/usr/bin/rt-setup-kdump
/usr/bin/slub_cpu_partial_off
/usr/sbin/kernel-is-rt

```

2.2. Installing Red Hat Enterprise Linux for Real Time Using Diskless Boot

Diskless booting allows a system administrator to deploy a large number of compute nodes without having to configure an equal number of local storage devices. This section provides instructions on how to set up a remote diskless system using an NFS filesystem mounted by a PXE booting client. Perform the following procedure on an existing installation of Red Hat Enterprise Linux for Real Time.

1. Install the following packages required to set up a basic remote diskless system booted over PXE:

```
~]# yum install tftp-server xinetd dhcp syslinux dracut-network
```



Important

- Ensure that the NFS and DHCP services are properly configured and started.
- Firewall rules which have been predefined by **system-config-firewall** can block **tftp** traffic and prevent the clients from booting. It is necessary to adjust the firewall rules on the server.

2. Enable the **tftp** service to allow PXE booting via the network. To do so, edit the `/etc/xinetd.d/tftp` file and set the **Disabled** parameter to **No**.
3. Copy `/usr/share/syslinux/pxelinux.0` to the **tftp** root directory in `/var/lib/tftpboot/`:

```
~]# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/
```

4. Create a `pxelinux.cfg` directory inside the **tftp** root directory:

```
~]# mkdir -p /var/lib/tftpboot/pxelinux.cfg/
```

5. Configure firewall rules to allow **tftp** traffic. Using a text editor, add the following line to `/etc/hosts.allow`:

```
tftpd: .hostname.com
```

Replace `.hostname.com` with the domain name of the target clients on which the installations will be performed. For more information on firewall rules refer to the man page for **hosts_access**.

- Restore the SELinux context on the **tftp** server. To provide the necessary files via the tftp protocol for PXE boot, run the following command:

```
~]# restorecon -R /var/lib/tftpboot
```

- To enable PXE booting on the DHCP server, add the following configuration to **/etc/dhcp/dhcpd.conf**:

```
allow booting;
allow bootp;
class "pxeclients" {
    match if substring(option vendor-class-identifier, 0, 9) =
    "PXEClient";
    next-server server-ip;
    filename "pxelinux.0";
}
```

Replace *server-ip* with the IP address of the host machine on which the **tftp** and DHCP services reside.

- The root directory of the exported file system (used by diskless clients in the network) is shared via NFS. Configure the NFS service to export the root directory by adding it to the **/etc/exports** file in the following format:

```
/exported/root/directory hostname.com(rw, sync, no_root_squash)
```

The example configuration exports the directory */export/root/directory* to all hosts on the *hostname.com* network with **read-write** permissions. The **sync** option ensures that the NFS server will not reply to requests before changes made by previous requests are written to disk. The **no_root_squash** option enables the NFS server to connect to the client with root privileges.

- To accommodate completely diskless clients, the root directory should contain a complete Red Hat Enterprise Linux for Real Time installation. You can synchronize this with a running system via **rsync**. Run:

```
~]# rsync -a -e ssh --exclude='/proc/*' --exclude='/sys/*'
hostname.com:/ /exported/root/directory
```

Replace *hostname.com* with the hostname of the running system with which to synchronize via **rsync**. The */exported/root/directory* is the path to the exported file system.

- Add the following configuration to the exported file system's **/etc/fstab** file:

```
none /tmp tmpfs defaults 0 0
tmpfs /dev/shm tmpfs defaults 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

- Select the kernel that diskless clients should use (**vmlinux-rt-kernel-version**) and copy it to the **tftpboot** directory:

```
~]# cp /boot/vmlinux-rt-kernel-version /var/lib/tftpboot/
```

12. Create the `initramfs` disk image (i.e. `initramfs-rt-kernel-version.img`) with network support:

```
~]# dracut -a "network nfs" initramfs-rt-kernel-version.img rt-
kernel-version
```

13. Copy the `initramfs` disk image to the `tftpboot` directory:

```
~]# cp initramfs-rt-kernel-version.img /var/lib/tftpboot/
```

14. Edit the default boot configuration to use the `initramfs` disk image and Realtime kernel inside `/var/lib/tftpboot`. This configuration instructs the diskless client to mount the exported file system (`/exported/root/directory`) as read-write. To do this, configure `/var/lib/tftpboot/pxelinux.cfg/default` with the following:

```
default realtime

label realtime
    kernel vmlinuz-rt-kernel-version
    append initrd=initramfs-rt-kernel-version.img root=nfs:server-
ip:/exported/root/directory rw
```

Replace `server-ip` with the IP address of the host machine on which the exported NFS root partition resides.

The NFS share is now ready for exporting to diskless clients. These clients can boot Red Hat Enterprise Linux for Real Time over the network via PXE.



Important

You must disable SELinux on diskless clients which use NFS as the root file system. Add **`SELINUX=disabled`** to the `/etc/selinux/config` file, or add **`selinux=0`** on the kernel command line when booting the client.

For more information on setting up remote diskless systems, see the [Storage Administration Guide](#).

2.3. Available RPM Packages

This section lists the RPM packages available in the repository for Red Hat Enterprise Linux for Real Time.

The **Red Hat Enterprise Linux for Real Time Specific** column indicates if the RPM differs from the standard Red Hat Enterprise Linux 7 maintenance stream, or is not applicable to the standard kernel. A **No** in this column indicates that the RPM performs equivalently on Red Hat Enterprise Linux 7.

The **Required** column indicates whether or not the package is mandatory for correct Red Hat Enterprise Linux for Real Time behavior. A **No** in this column indicates that usage is optional.

Table 2.1. Basic Red Hat Enterprise Linux for Real Time Kernel Packages

RPM Package Name	Description	Red Hat Enterprise Linux for Real Time Specific?	Required?
kernel-rt	Low latency and preemption functionality	Yes	Yes
rtctl	System start-up script used to configure the default Red Hat Enterprise Linux for Real Time scheduling priorities of kernel threads	Yes	Yes
kernel-rt-doc	Red Hat Enterprise Linux for Real Time documentation	Yes	Advised

The following packages contain test programs for use with Red Hat Enterprise Linux for Real Time.

Table 2.2. Red Hat Enterprise Linux for Real Time Development and Test Packages

RPM Package Name	Description
kernel-rt-devel	Headers and libraries for kernel development
kernel-rt-trace	Red Hat Enterprise Linux for Real Time kernel with tracing functions compiled in
kernel-rt-trace-devel	Headers and libraries for development on trace kernel
kernel-rt-debug	Red Hat Enterprise Linux for Real Time kernel with debugging functions compiled in (slow)
kernel-rt-debug-devel	Headers and libraries for development on debug kernel
rt-tests	Utilities for measuring system latencies and for proving that priority-inheritance mutexes function properly
rtcheck	A program that tests the running system for Red Hat Enterprise Linux for Real Time capabilities

The following set of packages are provided for use with **oprofile**, **systemtap**, and the **crash** utility for analyzing kernel crashdumps. The debugging packages consist of symbol tables and are quite large. For this reason, they are separately delivered from the other Red Hat Enterprise Linux for Real Time packages.

These packages can be found and downloaded from the **Red Hat Enterprise Linux for Real Time - Debug RPMs** repository.

Table 2.3. Red Hat Enterprise Linux for Real Time Debugging Packages

RPM Package Name	Description
kernel-rt-debuginfo	Symbols for profiling and debugging use, such as oprofile or systemtap
kernel-rt-trace-debuginfo	Symbols for profiling and tracing
kernel-rt-debug-debuginfo	Symbols for profiling and tracing

RPM Package Name	Description
<code>kernel-rt-debuginfo-common</code>	Common debuginfo files shared between the other debuginfo packages



Important

The packages in [Table 2.2, “Red Hat Enterprise Linux for Real Time Development and Test Packages”](#) and [Table 2.3, “Red Hat Enterprise Linux for Real Time Debugging Packages”](#) are not essential in order to run Red Hat Enterprise Linux for Real Time. They are provided as diagnostic tools only and are not normally needed. Using the `-trace` or `-debug` kernel packages negatively impacts performance and can render any benefit from the use of the realtime kernel negligible.

2.4. Post-Installation Instructions

Booting

The Red Hat Enterprise Linux for Real Time kernel is automatically set as the default boot kernel during the installation process. Therefore, you can start using Red Hat Enterprise Linux for Real Time when you reboot the computer.

If you ever want to boot the standard kernel, select it manually in the **GRUB** menu during booting. You can also make the standard kernel the default choice if you want. See [Changing the Default Boot Entry](#) in the System Administrator's Guide for details.

You can verify that your system is running the Red Hat Enterprise Linux for Real Time kernel by running the `uname` command:

```
~]# uname -a
Linux rt-server.example.com 3.10.0-release.rt56.build.el7.x86_64 ...
```

Check the output for the `rt` designation. If it appears, the Red Hat Enterprise Linux for Real Time kernel is running.

Kdump

You can configure Red Hat Enterprise Linux for Real Time to provide crash dump information by enabling `kexec/kdump`, if you have not done so for the standard kernel during the installation of Red Hat Enterprise Linux 7. Further information and instructions on how to configure your system to obtain kernel crash information can be found in the [Red Hat Enterprise Linux for Real Time Tuning Guide](#).

Differences Between Red Hat Enterprise Linux for Real Time and the Standard Kernel

Red Hat Enterprise Linux for Real Time differs substantially from the standard Red Hat Enterprise Linux 7 kernel. As a consequence, third-party kernel modules are incompatible with Red Hat Enterprise Linux for Real Time.

Kernel modules are inherently specific to the kernel they are built for. The realtime kernel is substantially different from the standard kernel, and so are the modules. In other words, you cannot take third-party modules from Red Hat Enterprise Linux 7 and use them as-is on Red Hat Enterprise Linux for Real Time. If you must use a third-party module, you must recompile it with the

Red Hat Enterprise Linux for Real Time header files, which are available in the *-devel* subpackage. See [Table 2.2, “Red Hat Enterprise Linux for Real Time Development and Test Packages”](#) for more information about all the subpackages.

The following are some example third-party drivers which ship for standard Red Hat Enterprise Linux 7 but do not currently have a custom build for Red Hat Enterprise Linux for Real Time:

- » EMC Powerpath
- » NVidia graphics
- » Advanced storage adapter configuration utilities from Qlogic

The user space **syscall** interface *is* compatible with Red Hat Enterprise Linux for Real Time.

Chapter 3. Red Hat Enterprise Linux for Real Time Kernel Tuning

The Red Hat Enterprise Linux for Real Time kernel offers many performance tuning parameters not otherwise available in Red Hat Enterprise Linux 7. In order to achieve optimal low-latency determinism it is necessary to perform Red Hat Enterprise Linux for Real Time specific system tuning.

Tuna

The primary diagnostic facility provided with the Red Hat Enterprise Linux for Real Time kernel is Tuna. Tuna provides both a command line tool and a graphical interface that can be used to change attributes of threads (scheduling policy, scheduler priority and processor affinity) and interrupts (processor affinity). The tool is designed to be used on a running system, and changes take place immediately. This allows any application-specific measurement tools to see and analyze system performance immediately after the changes have been made.

Diagnostic Tools

The Red Hat Enterprise Linux for Real Time kernel provides several diagnostic tools to test and report on latencies.

The **latency tracer** is a peak detector which is used to identify the longest running non-preemptable kernel codepaths. This is particularly useful for identifying whether non-deterministic performance results are attributable to the kernel or to user-space components. In customer deployments, the tool is most useful to differentiate whether delays are in the kernel or the application.

The **ftrace** utility is used to analyze and debug latency and performance issues that occur outside of user-space. It has a variety of options that allow you to use the utility in a number of different ways. It can be used to trace context switches, measure the time it takes for a high-priority task to wake up, measure the length of time for which interrupts are disabled, or list all the kernel functions executed during a given period.

Direct Memory Access with `rmem.ko`

Red Hat Enterprise Linux for Real Time includes a kernel module called `rmem.ko`. This module is not loaded by default, and is provided only to meet the realtime Java conformance tests - specifically the Technology Compatibility Kit (TCK) test in the Real Time Specification for Java's ([RTSJ](#)) conformance suite. It is an RTSJ requirement of this conformance test that Java programs have direct access to physical memory. This `/dev/rmem` capability allows user applications to map any arbitrary memory region.

The capability can only be enabled as a result of direct root system administrator action. To further prevent accidental usage, the capability is disabled unless `unprotected_address_space=1` is specified as a boot option for the module in the **GRUB** configuration. If the `rmem.ko` kernel module is explicitly loaded, the kernel will have a **tainted** flag. In this case, Red Hat Global Support may require you to reproduce any problems without the presence of `rmem.ko`.



Warning

Enabling `rmem.ko` gives any application direct access to physical memory. Many normal security mechanisms are bypassed and the system becomes much more vulnerable to attacks by malicious users. For this reason, it is strongly suggested that this feature be used *exclusively* for RTSJ certification purposes only. **DO NOT** use the `rmem.ko` module for production deployment.

There is another related boot option for this feature. This boot option reserves contiguous physical kernel memory at boot time for later usage by Java runtime. This memory is allocated at boot prior to the system memory becoming fragmented. The purpose of this is to avoid allocation failures for large contiguous memory. The line that would need to be added to the **GRUB** configuration is the following, where the *memsizes* parameter is expressed in bytes.

```
alloc_rtsj_mem.size=memsizes
```



Note

For comprehensive tuning information, see the [Red Hat Enterprise Linux for Real Time Tuning Guide](#).

Chapter 4. More Information

4.1. Reporting Bugs

Diagnosing a Bug

Before you file a bug report, follow these steps to diagnose where the problem has been introduced. This will greatly assist in rectifying the problem.

1. Check that you have the latest version of the Red Hat Enterprise Linux 7 kernel, then boot into it from the **GRUB** menu. Try reproducing the problem with the standard kernel. If the problem still occurs, report a bug against Red Hat Enterprise Linux 7.
2. If the problem does not occur when using the standard kernel, then the bug is probably the result of changes introduced in the Red Hat Enterprise Linux for Real Time specific enhancements Red Hat has applied on top of the baseline (3.10.0) kernel.

Reporting a Bug

If you have determined that the bug is specific to Red Hat Enterprise Linux for Real Time follow these instructions to enter a bug report:

1. Create a [Bugzilla](#) account if you do not have it yet.
2. Click on [Enter A New Bug Report](#). Log in if necessary.
3. Select the **Red Hat** classification.
4. Select the **Red Hat Enterprise Linux 7** product.
5. If it is a kernel issue, enter **kernel - rt** as the component. Otherwise, enter the name of the affected user-space component, such as **rteval**.
6. Continue to enter the bug information by giving a detailed problem description. When entering the problem description be sure to include details of whether you were able to reproduce the problem on the standard Red Hat Enterprise Linux 7 kernel.

Revision History

Revision 1-1	Thu Feb 19 2015	Radek Bíba
---------------------	------------------------	-------------------

Version for 7.1 GA release.