



Red Hat Enterprise Linux Hardware Certification 1.0

Test Suite User Guide

The Guide to Performing Red Hat Hardware Certification

Red Hat Enterprise Linux Hardware Certification 1.0 Test Suite User Guide

The Guide to Performing Red Hat Hardware Certification

Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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, OpenShift, 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

The Red Hat OpenStack Certification Test Suite User Guide explains the procedures necessary to certify hardware on Red Hat Enterprise software. It gives an overview of the entire certification process, then explains how to set up the certification environment, test the systems or components being certified, and submit the results to Red Hat for verification. Background information is also provided, including the test methodology and results evaluation.

Table of Contents

CHAPTER 1. INTRODUCTION	5
1.1. OVERVIEW	5
1.2. HARDWARE CERTIFICATION PROCESS SUMMARY	5
1.3. GIVING FEEDBACK AND GETTING HELP	6
CHAPTER 2. PREREQUISITES	8
2.1. PROGRAM MEMBERSHIP, ACCOUNTS, AND ENTITLEMENTS	8
2.2. IMPORTANT PRODUCT REQUIREMENTS	9
CHAPTER 3. PREPARING THE TEST ENVIRONMENT	10
3.1. TEST ENVIRONMENT OVERVIEW	10
3.2. PREPARE THE SYSTEM UNDER TEST (SUT)	11
3.3. PREPARE THE LOCAL TEST SERVER (LTS)	11
3.3.1. Hosting Prebuilt Guest Files on the LTS	11
3.4. PROXY SETTINGS FOR TEST SERVER AND TEST CLIENT	13
CHAPTER 4. CERTIFICATION WORKFLOW	15
4.1. CREATING RED HAT ENTERPRISE LINUX CERTIFICATION REQUESTS	15
4.1.1. Open a New System or Component Certification Request	15
4.1.2. Add Certifications to Previously Certified Hardware	15
4.1.3. Change Features or Hardware in an Existing Certification	15
4.2. CREATING AND PUBLISHING A SYSTEM CERTIFICATION	16
4.2.1. Copying an Existing System Certification to a New Entry	16
4.2.2. Creating a System Pass Through Certificate Using Existing Specification File	17
4.3. CREATING AND PUBLISHING A COMPONENT CERTIFICATION	17
4.3.1. Copying an Existing Component Certification to a New Entry	17
CHAPTER 5. LAYERED PRODUCT CERTIFICATIONS	19
5.1. LAYERED PRODUCT CERTIFICATIONS	19
5.1.1. Certifying for Red Hat OpenStack Platform Compute	20
5.1.2. Certifying for Red Hat Gluster Storage	20
5.1.3. Certifying for Red Hat Enterprise Linux for Real Time	20
5.1.4. Certifying Red Hat OpenStack Platform for Real Time Applications	21
CHAPTER 6. LEVERAGING	22
6.1. LEVERAGING	22
6.1.1. Rules for Leveraging from System Certification for Same Vendor	22
6.1.2. Rules for Leveraging from System Certification for Different Vendors	22
6.1.3. Generating Test Result ID for Leveraging from System Certification	23
6.1.4. Leveraging from Existing Component	23
CHAPTER 7. REGISTERING SUTS AND PREPARING TO TEST	25
7.1. OVERVIEW	25
7.2. REGISTER A SUT WITH THE LTS	25
7.3. ADD A PRODUCT TO THE LTS FROM THE RED HAT CERTIFICATION CATALOG	25
7.4. ADD A SANDBOX PRODUCT TO THE LTS	26
CHAPTER 8. RUNNING TESTS AND SUBMITTING LOGS FOR REVIEW	27
8.1. SELECTING AND RUNNING TESTS	27
8.2. VIEW THE TEST LOGS AND SUBMIT THE LOGS FOR REVIEW	27
8.3. RED HAT REVIEW OF TEST RESULTS	28
8.4. COMPLETING CERTIFICATIONS	29

APPENDIX A. REFERENCE MATERIAL	30
A.1. APPENDIXES	30
A.1.1. Hardware Test Procedures	30
A.1.1.1. 1GigEthernet	30
A.1.1.2. 10GigEthernet	30
A.1.1.3. 20GigEthernet	30
A.1.1.4. 25GigEthernet	31
A.1.1.5. 40GigEthernet	31
A.1.1.6. 50GigEthernet	31
A.1.1.7. 100GigEthernet	31
A.1.1.8. audio	32
A.1.1.9. battery	33
A.1.1.10. bluray	34
A.1.1.11. cdrom	38
A.1.1.12. core	38
A.1.1.13. cpuscaling	39
A.1.1.14. dvd	41
A.1.1.15. Ethernet	41
A.1.1.16. expresscard	42
A.1.1.17. fv_core	43
A.1.1.18. fv_memory	43
A.1.1.19. fv_network (Optional for SR-IOV)	43
A.1.1.20. fv_storage (Optional)	43
A.1.1.21. infiniband connection	43
A.1.1.22. info	44
A.1.1.23. iwarp connection	45
A.1.1.24. kdump	46
A.1.1.25. lid	46
A.1.1.26. memory	47
A.1.1.27. network	48
A.1.1.28. omnipath connection	50
A.1.1.29. pccard (Red Hat Enterprise Linux 6 only)	51
A.1.1.30. profiler	51
A.1.1.31. realtime	51
A.1.1.32. reboot (Optional)	52
A.1.1.33. RoCE connection	53
A.1.1.34. SATA	53
A.1.1.35. SATA_SSD	54
A.1.1.36. M2_SATA	54
A.1.1.37. U2_SATA	55
A.1.1.38. SAS	55
A.1.1.39. SAS_SSD	56
A.1.1.40. PCIE_NVMe	56
A.1.1.41. M2_NVMe	56
A.1.1.42. U2_NVMe	57
A.1.1.43. NVDIMM	57
A.1.1.44. STORAGE	58
A.1.1.45. suspend (Laptops only)	59
A.1.1.46. tape	61
A.1.1.47. USB2	61
A.1.1.48. USB3	61
A.1.1.49. video	62
A.1.1.50. WirelessG	63

A.1.1.51. WirelessN	63
A.1.1.52. WirelessAC (Red Hat Enterprise Linux 7 only)	63
A.1.2. Manually Adding Tests	63

CHAPTER 1. INTRODUCTION

1.1. OVERVIEW

The Red Hat Certification Program is Red Hat's method of certifying hardware and software to be compatible with Red Hat Enterprise Linux, Red Hat OpenStack Platform, Red Hat Gluster Storage, Red Hat Enterprise Linux for Real Time, and other Red Hat software products. The program has three main elements: the test suite, which tests the hardware or software undergoing certification; the [Red Hat Certification Catalog](#), which displays certified hardware and software to the public; and a joint support relationship between Red Hat and the vendor whose hardware or software is undergoing certification.

This workflow guide covers all aspects of the hardware portion of the Red Hat Certification process, from the initial request for certification to the final approval and posting of the certification. It is geared towards the engineer who has been tasked with certifying their company's products to run one or more of Red Hat's products. It assumes that you as the tester have a basic level of knowledge about the software that is being used for certification, and are familiar with concepts like operating system installation, software installation, and where applicable, hardware installation and removal.

The policies and other rules of the program are covered in the [Red Hat Hardware Certification Policy Guide](#).

1.2. HARDWARE CERTIFICATION PROCESS SUMMARY

Hardware certification covers the testing of servers, desktops/workstations, laptops, and individual components to run Red Hat Enterprise Linux, Red Hat OpenStack Platform Compute, Red Hat Gluster Storage, and Red Hat Enterprise Linux for Real Time.

Preliminary Steps

1. The partner establishes a certification relationship with Red Hat.
2. The partner stands up a test environment consisting of the partner's product and the Red Hat product combination to be certified.
3. The partner does preliminary testing to ensure this combination works well.
4. The partner installs the redhat-certification tool.

Certification Steps:

1. The partner creates a certification request for a specific system or hardware component using redhat-certification.
2. Red Hat's certification team applies the certification policies to the hardware specifications to create the official test plan.
3. The partner runs the tests specified in the official test plan and submits results using redhat-certification to Red Hat for analysis.
4. The certification team analyzes the test results and marking credit as appropriate and communicating any required retesting.
5. The partner provides Red Hat with a representative hardware sample that covers the items that are being certified.

6. When all tests have passing results, the certification is complete and the entry is made visible to the public on the external Red Hat Hardware Certification website at <https://access.redhat.com/certifications>.

The preliminary steps are further explained below in [Chapter 2, *Prerequisites*](#) and [Chapter 3, *Preparing the Test Environment*](#), and the certification steps are expanded upon in [Chapter 4, *Certification Workflow*](#).

If you need help at any time, refer to [Section 1.3, “Giving Feedback and Getting Help”](#), which explains how to contact Red Hat Certification for assistance.

1.3. GIVING FEEDBACK AND GETTING HELP

Partners who have a dedicated support resource that is an assigned Engineering Partner Manager, Engineering Account Manager, or Technical Account Manager can open a support case using the same tool they use to request support for other Red Hat products.

Partners who do not have a dedicated support resource can open a support case using [Red Hat Customer Portal](#) under the following instances:

- To report issues and get help with the certification process
- To submit feedback and request enhancements in the certification toolset and documentation
- To receive assistance on the Red Hat product on which your product or application is being certified. To receive Red Hat product assistance, it is mandatory to have the required product entitlements and subscriptions which are separate from certification-specific entitlements and subscriptions

To open a support case using Red Hat Customer Portal Interface, complete the following steps:

1. Log in to the [Red Hat Customer Portal](#) using the Red Hat account credentials that are also used to access other Red Hat assets like [Red Hat Connect for Technology Partners](#) and software subscriptions.
2. Click **Open a Support Case** on the Red Hat Customer Portal Home Page.
3. Complete the Support Case Form with special attention to the following fields:
 - From the **Product** field, select the name of the Red Hat product on which your product/application is being certified, based on the following details:
 - For Red Hat OpenStack Platform Certification, select **Red Hat OpenStack Platform**.
 - For Certified Cloud and Service Provider (CCSP) Certification, select **Red Hat Enterprise Linux**.
 - For Red Hat Container Certification, select **Red Hat Enterprise Linux**.
 - For Red Hat Hardware Certification, select **Red Hat Enterprise Linux**.
 - From the **Product Version** field, select the version of the product.
 - In the **Problem Statement** field, type a problem statement/issue or feedback using the following format: **{Partner Certification} (The Issue/Problem or Feedback)**. Replace **(The Issue/Problem or Feedback)** with either the issue/problem faced in the certification process/Red Hat product or feedback on the certification toolset/documentation.

For example: {Partner Certification} Error occurred while submitting certification test results using Red Hat Certification application

Complete the remaining form using the details <https://access.redhat.com/articles/38363>.

CHAPTER 2. PREREQUISITES

2.1. PROGRAM MEMBERSHIP, ACCOUNTS, AND ENTITLEMENTS

Red Hat requires a relationship agreement that is specific to the type of certification being performed before we can accept certification requests. This relationship can be documented in a section or subsection of an OEM or other partnership agreement, or it can be established by an independent certification agreement. The creation of an OEM or other partnership agreement is not covered here, as that is something handled by other groups. Talk with your assigned Red Hat representative if you want to know more about partnerships beyond hardware certification. If you would like to establish an independent hardware certification agreement, you can do so using the following steps:

1. Purchase membership in the Red Hat Certification Program.
Email your sales contact and ask to purchase membership in the program. If you do not have a sales contact at Red Hat, email your technical account manager, partner manager, or send an email to cert-ops@redhat.com for assistance purchasing program membership. Please provide the following information:
 - a. Text indicating that you wish to purchase membership in the Red Hat Hardware Certification Program
 - b. Contact information for the person who will be placing the order
 - c. Your company's legal name
 - d. Billing address (Please **do not** include credit card information.)
A Red Hat sales representative will contact you to complete the purchasing process, which includes the creation of an account on the [Red Hat Customer Portal]<https://access.redhat.com>).
2. Create a Red Hat Customer Portal account and login.
 - a. Navigate to Red Hat Customer Portal.
 - b. Select Register in the menu and follow the instructions.
Make sure that you create a company account and not a personal account. The account created during this step is also used to sign in to the Red Hat Hardware Catalog when working with certification requests.



NOTE

The certification test suite uses Red Hat Customer Portal single sign-on (SSO) credentials to log in to the Red Hat Certification site and the Red Hat Certification test suite. If you already have a Customer Portal login skip this step. If you do not have a login but your company has logins on the Portal, please ask your company's organization administrator to create a login for you under the company's account.



NOTE

A Red Hat Customer Portal organization administrator in your organization will have the permissions to create an SSO account for your organization. If you are not familiar with the Red Hat Customer Portal organization administrator for your organization, try contacting your engineering account manager or engineering partner manager for assistance.

1. Obtain required Red Hat Certification Catalog permissions.

- a. File a support request to have certification creation permissions added to your Red Hat Customer Portal login.

If you have an assigned technical account manager (TAM) or engineering partner manager (EPM), file a request through Red Hat Bugzilla at <http://bugzilla.redhat.com>.

If you do not have a TAM, file a request through Red Hat's Customer Portal at <https://access.redhat.com>. Include your Red Hat Customer Portal account user name in the request.

After this request has been approved and you log in to the Red Hat certification(rh-cert) web user interface, the Create link will appear in the rh-cert menu. This allows you to create a new certification request.



NOTE

It is mandatory to obtain Red Hat Certification permissions on your Red Hat Customer Portal account before it can be used for certification purposes.

2.2. IMPORTANT PRODUCT REQUIREMENTS

Hardware certification must be performed using drivers provided by Red Hat. Any hardware requiring third-party drivers for enablement is not eligible for certification.

CHAPTER 3. PREPARING THE TEST ENVIRONMENT

3.1. TEST ENVIRONMENT OVERVIEW

The test environment for Red Hat Enterprise Linux certification consists of at least two networked machines, each running Red Hat Enterprise Linux and the certification test suite. The first machine is the system under test (**SUT**) and contains the hardware that will undergo certification. The SUT runs the version of Red Hat Enterprise Linux on which the hardware is being certified.

The second machine is the local test server (**LTS**) which serves as the command and control unit that issues test commands to the SUT. The LTS runs the latest version of Red Hat Enterprise Linux 7.x. A single LTS system can control multiple SUT systems, but it should only perform network or kdump testing on one SUT at a time due to network bandwidth limitations.

The appropriate version(s) of Red Hat Enterprise Linux for your environment can be downloaded from the Red Hat Customer Portal at: <https://access.redhat.com/downloads/content/69>.

The certification test suite that runs on both the LTS and SUT is composed of the following packages:

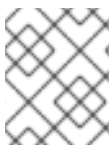
- dt
- lmbench
- stress
- redhat-certification-backend
- redhat-certification-hardware
- redhat-certification (RHEL 7 only)
- python-django (RHEL 7 only)
- python-django-bash-completion (RHEL 7 only)

All test suite packages are available on the Red Hat Customer Portal at: <https://access.redhat.com/downloads/content/282/>.

The certification test suite also requires two debuginfo files to be installed on the SUT:

- **kernel-debuginfo-\$VERSION**
where **\$VERSION** is the running kernel version number as shown in the output of **uname -a** or via the kernel RPM filename
- **kernel-debuginfo-common-\$ARCH-\$VERSION**
where **\$ARCH** is x86_64, i686, etc. and **\$VERSION** is the same as above

The debuginfo packages can be downloaded from the Red Hat Customer Portal as explained in the Appendix of the following Red Hat Knowledgebase article: <https://access.redhat.com/solutions/9907>.



NOTE

Be sure to follow the directions in the Appendix for downloading the files; do not register the SUT with the Red Hat Customer Portal and download using that method.

3.2. PREPARE THE SYSTEM UNDER TEST (SUT)

1. Locate the SUT and all the components that must be tested as part of the certification activities.
2. Download the appropriate architecture and version of RHEL, the certification test suite, and the necessary debuginfo packages from the locations mentioned earlier and install them on the SUT.

The OS should be configured as explained in the appropriate RHEL kickstart file that can be found using the following link. Choose the file that matches the version and architecture of RHEL you are certifying: <http://people.redhat.com/gcase/rhcert-2/ks/>.

If you are not using kickstart to perform your installation, please consult the guide at the top of the kickstart file for more information on proper, manual installation.

3.3. PREPARE THE LOCAL TEST SERVER (LTS)

1. Locate an appropriate machine to function as the LTS.
The LTS is not required to be a certified system; however, to ensure proper functionality, network connectivity should be of equal or greater speed than the interfaces on the SUT in order to properly test the SUT's network devices.
2. Download the appropriate architecture and version of
 - Red Hat Enterprise Linux: <https://access.redhat.com/downloads/content/69>
 - Certification Test Suite packages: <https://access.redhat.com/downloads/content/282/>
 - debuginfo packages: <https://access.redhat.com/solutions/9907>
3. Install them on the LTS.
4. Configure the OS as explained in the appropriate RHEL kickstart file available at <http://people.redhat.com/gcase/rhcert-2/ks/>. Choose the file for RHEL 7.
If you are not using kickstart to perform your installation, consult the guide at the top of the kickstart file for more information on proper manual installation.
5. Run the following command to start Apache, Red Hat Certification back-end server and the server listener process: **# rhcertd start**

3.3.1. Hosting Prebuilt Guest Files on the LTS

If the system being tested supports virtualization, that feature must be tested. We have prebuilt guest files that are automatically downloaded by the SUT during the `fv_*` tests to satisfy this requirement. Those files can be hosted on your LTS if you wish to shorten the download time from the Red Hat FTP site, or if your testing environment is disconnected from the network.

The files for **x86_64** architecture are available at the following location:

- <ftp://partners.redhat.com/a166eabc5cf5df158922f9b06e5e7b21/hwcert/fv-images/RHEL7/>
- <ftp://partners.redhat.com/a166eabc5cf5df158922f9b06e5e7b21/hwcert/fv-images/RHEL6/>

The files for **aarch64 fv testing (RHEL7 only)** are available at the following location:

- <ftp://partners.redhat.com/a166eabc5cf5df158922f9b06e5e7b21/hwcert/fv-images/RHEL7/aarch64/>

The files for **ppc64le fv testing (RHEL7 only)** are available at the following location:

- <ftp://partners.redhat.com/a166eabc5cf5df158922f9b06e5e7b21/hwcert/fv-images/RHEL7/ppc64le/>

The files for **s390x(alt)** are available at the following location:

- <ftp://partners.redhat.com/a166eabc5cf5df158922f9b06e5e7b21/hwcert/fv-images/RHEL7/s390x/>

The kickstart files mentioned earlier have a section in them to automatically download and install the files on the LTS. If you wish to manually download the files and place them on your LTS, here are the steps to take:

After setting up a local test server as explained in section [Prepare the Local Test Server](#).

Certify for Red Hat Enterprise Linux 7

1. Create a directory on the LTS called **/var/www/rhcert/store/transfer/fv-images/RHEL7**.
2. Copy the following files of Red Hat Enterprise Linux 7 FTP site to the local directory:

x86_64 architecture files:

- **hwcertData.img.tar.bz2** Results transfer package from guest to host
- **hwcert-x86_64.xml.tar.bz2** Full-virt guest configuration file for x86_64
- **hwcert-x86_64.img.tar.bz2** Full-virt KVM guest image for x86_64

aarch64 fv testing files:

- **hwcertData.img.tar.bz2** Results transfer package from guest to host
- **hwcert-aarch64.xml.tar.bz2** Full-virt guest configuration file
- **hwcert-aarch64.img.tar.bz2** Full-virt KVM guest image
- **hwcert-aarch64_VARS.fd.tar.bz2** Full-virt nvram file

ppc64le fv testing files:

- **hwcertData.img.tar.bz2** Results transfer package from guest to host
- **hwcert-ppc64le.xml.tar.bz2** Full-virt guest configuration file
- **hwcert-ppc64le.img.tar.bz2** Full-virt KVM guest image

s390x(alt) files:

- **hwcertData.img.tar.bz2** Results transfer package from guest to host
- **hwcert-s390x.xml.tar.bz2** Full-virt guest configuration file
- **hwcert-s390x.img.tar.bz2** Full-virt KVM guest image

Certify for Red Hat Enterprise Linux 6

Setting up Red Hat Enterprise Linux 6

1. Create a directory on the LTS called `/var/www/rhcert/store/transfer/fv-images/RHEL6`.
2. Copy the following files from the Red Hat Enterprise Linux 6 FTP site above into the local directory:
 - `hwcertData.img.tar.bz2` Results transfer package from guest to host
 - `hwcert-x86_64.xml.tar.bz2` Full-virt guest configuration file for x86_64
 - `hwcert-x86_64.img.tar.bz2` Full-virt KVM guest image for x86_64

NOTE

If you are using redhat-certification version redhat-certification-5.3-20171023.6.el7 or later on the LTS, and redhat-certification-hardware version prior to redhat-certification-hardware-5.6 on the SUT, you must first add the following to a new file on the LTS, `/etc/httpd/conf.d/atemp.conf`:

```
Alias /store "/var/www/rhcert/store"
<Directory "/var/www/rhcert/store">
    Options Indexes FollowSymlinks
    Order allow,deny
    Allow from all
</Directory>
```

Make certain to run `rhcertd restart` on the LTS after saving the file.

3.4. PROXY SETTINGS FOR TEST SERVER AND TEST CLIENT

If your network utilizes a proxy, you may need to manually configure the test server and/or test client for the proxy as outlined below:

In the test server, update the `/etc/rhcert.xml` file as per the following settings:

```
<urls>
<proxy-url protocol="http">PROXY_SERVER:PROXY_PORT</proxy-url>
<proxy-url protocol="https">PROXY_SERVER:PROXY_PORT</proxy-url>
</urls>
```

Replace **PROXY_SERVER** with the IP or dns-name of your proxy server, and **PROXY_PORT** with your proxy port number.

For example:

```
<proxy-url protocol="http">http://rhcert-example.redhat.com:3148</proxy-
url>
<proxy-url protocol="https">https://rhcert-example.redhat.com:3148</proxy-
url>
```

To open port 80 and port 8009 on test server and test client, run the `rhcert-cli register` command.

For more information, we recommend you to refer [How can we access to the Hardware Certification \(rhcetd web interface\) via proxy?](#)

CHAPTER 4. CERTIFICATION WORKFLOW

4.1. CREATING RED HAT ENTERPRISE LINUX CERTIFICATION REQUESTS

4.1.1. Open a New System or Component Certification Request

To create a new certification request, complete the following steps:

1. In your test server, launch **Red Hat Certification** web user interface in a browser using the `http://<machine-IP>` link.
2. Type your **Red Hat account credentials** previously enabled for certification in the **Username** and **Password** fields. Click **Login**.
3. Click the **New Certification** button. This will take you to **Choose the Red Hat Certification** web page.
4. From the **Product** drop-down list, select **Red Hat Enterprise Linux**. The **Version** and **Platform** value gets generated automatically. However, partners can select the version and platform fields according to their requirement. Click **Next**. This will take you to **Choose the product to be certified** web page.
5. Select the **Vendor**, **Make**, and **Name** items from the drop-down list. Click **Next**.
6. A notification of the requested Hardware certification for the new product is displayed.
7. To publish the certificate and change the status from **In Progress** to **Passed**, click the **Publish** tab, check the option **Publish this certification on** and from the drop down list, select the appropriate Year, Month, and Day. Click **Save**.

After the request is created, monitor the request for questions from the review team as they create the official test plan.

Testing can begin as described in section [Layered Product Certifications](#) when the test plan is complete.

4.1.2. Add Certifications to Previously Certified Hardware

This process is used, for example, when creating a certification request for RHEL 7 on a system that already has an in-progress or completed RHEL 6 certification request.

1. Log in to Red Hat Certification and click the **New Certification** button.
2. Select the Red Hat product, version, and platform for certification and click **Next**.
3. Select a vendor, make, and name of the product to be certified from the dropdown lists and click **Next**.

After the request is created, monitor the request for questions from the review team as they create the official test plan. Testing can begin as described in section [Layered Product Certifications](#) when the test plan is complete.

4.1.3. Change Features or Hardware in an Existing Certification

A supplemental certification is used when new hardware or features are added to an existing certified system or component. To create supplemental certification Red Hat certification team will initially add the supplemental components to the specification file. Perform the following steps to attach the specification file:

1. Click on the existing hardware cert this will take you to the hardware cert web page that have sections **Product** and **Certification**.
2. Click **Product**.
3. Click the **Product Details** tab.
4. In the **Attachment** field, click **Browse** to attach the specification file for the new supplemental component. Select the checkbox **is this a specification**.
Once the Red Hat certification team adds the supplemental components based on the specification file, perform the following steps to **create** the Supplemental certification:
5. Go to the hardware cert web page that have two sections **Product** and **Certification**.
6. Click **Certification**.
7. Click the **Related Certification** tab for creating supplemental certification.
Red Hat certification team adds the test plan to the newly generated certification.
8. In the **Related Certification** tab go to the **Supplemental Certifications** section click the **New Certification** button to create a new Supplemental certification.

After the supplemental certification is successfully created, Partners can start testing on it.

4.2. CREATING AND PUBLISHING A SYSTEM CERTIFICATION

A system pass-through certification essentially creates a copy of a certified system, listing it under a different vendor name, a different make, or a different model.

Pass-through is used when a vendor sells their system to a partner who then rebrands it, or if a vendor sells two or more systems where one system is a superset of the other(s). In such situation, the vendor should have tested the existing system certification as that covers all the hardware in the new Pass through certification.

1. To create a system certification refer Steps 1 to 4 of [Open a New System or Component Certification Request](#)
2. Select the **Vendor**, **Make**, and **Name**. Click the **New Product** button. This will take you to **Choose the Certification Program** web page.
3. Select the **Vendor** and **Program** as **Hardware**. Click the **Next** button. This will take you to **Define the Red Hat Hardware Certification Vendor Product** web page.
4. Fill in all the relevant details. From the drop down list of the **Category** field, select the category as **System**.

This creates the System Certification. The Red Hat certification team certifies and publishes the newly created System Certification. After the certificate is certified and published, it becomes public for other partners to refer it as a pass through component.

4.2.1. Copying an Existing System Certification to a New Entry

1. To create the Pass through certification, go to the **Red Hat certification** web user interface, click the existing hardware system certification that is certified. Click the **Certification** Section. In the, **Related Certification** tab, go to the **Pass through Certification** section and click the **New Certification** button.
2. In the **Vendor** field select the Vendor whose product you need to pass-through. In the **Make** field select the make that you need to pass through.
3. Click the **Create** button. This will generate a request to create a pass through system specification and a pass through certification for the generated specification.

If the original system specifications and the pass-through system specifications are identical or have no differences, no additional testing will be required. If differences are found, the Red Hat certification team will discuss with you what should be done to account for them.

4.2.2. Creating a System Pass Through Certificate Using Existing Specification File

Following are the steps to create a system pass through certificate for a new model using the existing model specification file:

1. Go to the **Red Hat certification** web user interface, click the existing hardware system certification that is certified. Click the **Certification** Section.
2. In the **Related Certification** tab, go to the **Pass through Certification** section and chose the pass through specification file that has been created.

This will create the second pass-through certificate using the same specification entry.

4.3. CREATING AND PUBLISHING A COMPONENT CERTIFICATION

A component pass-through certification essentially creates a copy of a certified component, listing it under a different vendor name, a different make, or a different model. This type of pass-through is used when a system vendor wants to include a component that has already been certified by a component vendor, when a component vendor sells their components to a third party who rebrands them, or if a vendor sells two or more components where one system is a superset of the other(s).

1. To create a component certification refer Steps 1 to 4 [Open a New System or Component Certification Request](#)
2. Select the **Vendor**, **Make** and **Name**. Click the **New Product** button. This will take you to **Choose the Certification Program** web page.
3. Select the **Vendor** and **Program** as Hardware. Click the **Next** button. This will take you to **Define the Red Hat Hardware Certification Vendor Product** web page.
4. Fill in all the relevant details. From the drop down list of **Category**, select the category as **Component/Peripheral**.

This creates the Component certification. The Red Hat certification team certifies and publishes the newly created Component certification. After the certificate is certified and published, it becomes public for other partners to refer it as a pass through component.

4.3.1. Copying an Existing Component Certification to a New Entry

Following are the steps to copy an existing component certification to a New Entry:

1. To copy the Component certification, go to the **Red Hat Certification** web user interface, click the existing hardware system certification that is certified. Click the **Certification** section. In the **Related Certification** tab, go to the **Pass through Certification** section and click the **New Certification** button.
2. In the **Vendor** field select the Component Vendor whose product you need to pass-through. In the **Make** field select the Component Make that you need to pass through.

**NOTE**

Here, the Component Vendor and the Component Make are the fields that gets generated while performing Steps 1 to 4 of [Creating and Publishing a Component Certification](#).

If the original component specifications and the pass-through component specifications are identical then, no additional testing will be required. If there are differences found, the Red Hat certification team will discuss with you what should be done to account for them.

CHAPTER 5. LAYERED PRODUCT CERTIFICATIONS

5.1. LAYERED PRODUCT CERTIFICATIONS

Layered product certifications are built up on a completed Red Hat Enterprise Linux certification and list a system as **Certified** for additional Red Hat software products. At this time, we offer layered product certifications for Red Hat OpenStack Platform Compute, Red Hat Gluster Server, and Red Hat Enterprise Linux for Real Time.

Partners can create Layered Product Certifications either by **Automatic generation of a Layered Cert** or **Manual creation of a Layered Cert**

- **Generating a Layered Cert Automatically:**

A layered cert can be generated automatically only if the status of the hardware cert is **Certified**. The Red Hat certification team makes the hardware cert **Public** for it to be listed on the Red Hat Certification web user interface. Following are the steps to auto generate a hardware layered cert:

1. Click on the hardware cert that has to be certified and made public on the **Red Hat Certification** web user interface.
2. Click the **Dialog** tab.
3. In the **New Comment** text box enter the comment (for example, "Please certify and make the cert public").
4. Click the **Add Comment** button.

After the Partner adds comment for the requested hardware cert, the Red Hat certification team certifies and makes the certificate public. Partners will receive an email once the cert is certified and public.

Login to the Red Hat Certification web user interface. The new auto created layered certs will be listed on the web user interface. If not, click the Refresh button, the new certs should be downloaded.

- **Creating a Layered Cert Manually:**

A layered cert can be manually created for hardware certs that has the status **In Progress**. Following are the steps to manually create the layered cert:

1. Click on the hardware cert that has to be certified from the **Red Hat Certification** web user interface.
2. Click the **Dialog** tab.
3. In the **New Comment** text box enter the comment for the Red Hat certification team to certify the hardware cert. For example "Please certify the hardware cert".
4. Click the **Add Comment** button.

After the request is completed by the Red Hat certification team, Partners can manually create the layered cert from the **Red Hat Certification** web user interface in either of the following two ways:

- a. **Selecting the Hardware Cert:**

1. From the **Red Hat Certification** web user interface, click the hardware cert that is certified from the Red Hat certification and needs Layered cert.

2. Click the **Related Certification** tab.
3. In Layered Certifications section, click the **New Certification** button. This will take you to the **New Layered Certification** web page.
4. Select the **Product** and **Version** from the drop down list.
5. Click **Create** button.

b. **Using the New Certification button:**

1. From the **Red Hat Certification** web user interface, click the **New Certification** button. This will take you to **Choose the Red Hat certification** web page.
2. From the **Product** drop-down list, select the **Layered Product**. The **Version** and **Platform** value gets generated automatically. However, partners can select the version and platform fields according to their requirement. Click **Next** button. This will take you to **Choose the product to be certified** web page.
3. Select the **Vendor**, **Make**, and **Name** items from the drop-down list. Click **Next** button.

A notification of the requested Hardware certification for the new product is displayed. The newly created layered cert will be visible on the Red Hat Certification web user interface.

5.1.1. Certifying for Red Hat OpenStack Platform Compute

A certification entry for Red Hat OpenStack Platform Compute is automatically created for every Red Hat Enterprise Linux 7 x86_64 server certification request. No additional tests beyond Red Hat Enterprise Linux certification are needed to achieve the Red Hat OpenStack Platform Compute certification. Your system will be marked **Certified** for both Red Hat Enterprise Linux and Red Hat OpenStack Platform Compute on successful completion of the Red Hat Enterprise Linux certification, assuming no issues are present that would cause problems running the OpenStack Platform Compute node. If you have any questions about this, please contact your Red Hat Support representative.

5.1.2. Certifying for Red Hat Gluster Storage

A certification entry for Red Hat Storage Server is automatically created for every Red Hat Enterprise Linux 6, and 7 x86_64 server certification request.

No additional testing beyond the normal Red Hat Enterprise Linux certification tests are required to receive this certification, but a specification review will be performed by the review team before the certification can be granted.

More information on the specification requirements for Red Hat Storage Server can be found at the Red Hat Knowledgebase article entitled [Red Hat Storage Server 2.1 Compatible Physical, Virtual Server and Client OS Platforms](#).

5.1.3. Certifying for Red Hat Enterprise Linux for Real Time

Red Hat Enterprise Linux for Real Time is used for time-critical workloads that need to execute in a defined, predictable way.

Any server that is certified to run Red Hat Enterprise Linux 7, AMD64 and Intel 64 architecture, is eligible to be certified for Real Time.

If you are interested in certifying a system to run Real Time, confirm that it is already certified for Red Hat Enterprise Linux 7, x86_64 architecture, then talk to your Red Hat Support representative to learn more about Real Time.

5.1.4. Certifying Red Hat OpenStack Platform for Real Time Applications

Red Hat Openstack Platform for Real-Time Applications is designed to deliver ultra-low latency for performance-sensitive environments.

Red Hat OpenStack Platform for Real-Time Applications Certification, relatively verifies that the cyclic test performed on a VM using a dedicated pinned CPU does not exceed a maximum latency as defined in the [Red Hat Hardware Program Policy Guide](#).

Partners are expected to perform the `fv_real-time` test when the kernel Real-Time (`kernel-rt`) is running and the supported full virtualization is enabled by the machine. If the SUT is not connected to the internet, Partner will need to download the `qcow2` image to the LTS. Run the following commands as root to download the `qcow2` image:



NOTE

See the section [Hosting Prebuilt Guest Files on the LTS](#) for *partner ftp URL*.

```
cd /var/www/rhcert/
mkdir -p store/transfer/fv-images/RHEL7/
cd store/transfer/fv-images/RHEL7/
wget <partner ftp url>/hwcert/fv-images/RHEL7/rhel-kvm-rt-
image.qcow2.tar.bz2
cd -
chown -R --reference=store store
rhcertd restart
```



IMPORTANT

While the test is running, the connection to the SUT may be lost as the machine will reboot automatically for the configuration changes performed by the test to take effect. This test takes over 12 hours to complete and maximum latency could be 40 microsecond or less.

CHAPTER 6. LEVERAGING

6.1. LEVERAGING

Leveraging is the reuse of passing test results from hardware in a certified system to cover testing of identical hardware in a new certification request. It can only be used for certain optional items, and these items must be identical. You cannot leverage test results for a new model component with the test results from an old model, no matter how similar they are; the items must be an exact match. Furthermore, leveraging can only be used on tests that **your organization or its agent** has performed.

6.1.1. Rules for Leveraging from System Certification for Same Vendor

Following are the guidelines that should be taken care of while performing leveraging from system or component certification in case of same vendor:

1. Components must be identical.
2. Results generated must be from hardware of identical architecture.
3. System leveraging component results must certify the same major release.
4. Cross-vendor leveraging can not be performed for leveraging from system certification.

For example,

- Acme Computers can leverage the passing test results from any of its component to cover the identical item of another Acme system

But,

- Acme Computers cannot refer the test results from certifications performed by the Cloverleaf Industries

6.1.2. Rules for Leveraging from System Certification for Different Vendors

1. In a scenario where a component manufacturer creates pass-through of original certification using Vendor, Make, and Model information of the component as sold by the reseller the following guidelines should be considered:
 - i. In the **Advanced** tab select **Create using Pass-Through** of the original certification, just like the system pass-through certification
 - ii. If many resellers are using the same component, component manufacturer should create one pass-through for each reseller
 - iii. If a reseller uses multiple names for the same card, component manufacturer should create one pass-through for each name
2. After the Red Hat certification team confirms the hardware used are identical and the pass-through certification is complete by using the specification file documentation, the certification will be published or unpublished.
3. Reseller should use **certification ID** of their pass-through in appropriate **Leverage** field of their system certification requests that contain this hardware.

6.1.3. Generating Test Result ID for Leveraging from System Certification

Following are the steps for generating a test result ID for leveraging from system certification:

1. Create the source hardware product and certification from Red Hat Certification web user interface. Refer steps 1 to 7 of Section [Open a New System or Component Certification Request](#)
2. To add components to the newly created certification, from the **Red Hat Certification** web user interface, click the hardware cert that is certified. Click on the **Product** section and click the **Product Details** tab.
3. In the **Attachments** section, click the **Choose File** button to upload the specification file. The specification file consists of the component(s) that needs to be added.
4. Select the **is this a specification** checkbox, and add a brief note in the **Attachment Description** textbox. For example, "this is a spec.file".
5. From the **Red Hat Certification** web user interface, click the hardware cert that is certified. Click the **Certification** Section. In the **Progress** tab you will see the test plan is generated with respect to the components.
The Red Hat certification team reviews and adds the components mentioned in the specification file, and later creates a test plan for the added components.
6. Click the **Run** button to run tests for the components shown in the table. This will take you to the **Testing** tab.
7. Click **Add Test System**. This will take you **Select Host** web page.
8. Select the host for which you want the test to run and click the **Test** button.
9. In the **Testing** tab, click the **Continue Testing** button, this will generate the list of components.
10. Select the components for which you want to run the test and click the **Run Selected** button.
11. Once the test run is completed, you will get the message **Finished test run**.
12. Click on the test, the components on which the tests were run will have the results as **PASS**. To submit the test results to Red Hat certification team, from **Actions** field select **Submit** from the drop down list. This will take you to the **Submitting File** web page.
13. Click the **Submit** button.
The Red Hat certification team approves the test results. The approved test results generates a **test result id** that is associated with the component.
14. From the **Red Hat Certification** web user interface, click the hardware cert that is **certified**. Click the **Certification** Section. In the **Progress** tab, you will see the **Test Plan Credit** as **Confirmed**. The **Test Result** column will show the generated **Test Result ID**.

6.1.4. Leveraging from Existing Component

If you want to create a new certification using the same components, you can leverage that component. Following are the steps to leverage from an existing component:

1. Refer steps 1 to 4 of Section 6.1.3 [Generating Test Result ID for Leveraging from System Certification](#).
The Red Hat Certification team approves the components to leverage mentioned in the specification file.

2. From the **Red Hat Certification** web user interface, click the hardware cert that is **certified** and whose test result id you will leverage. Click the **Certification** Section. In the **Progress** tab, go to the **Test Result** column that shows the generated **Test Result ID** of the component that you will leverage. Copy the test result id by selecting the ID.
3. If the ID is successfully copied, you will get a message “**Copied Leverage Information from System Certification<the_component_name>**”.
4. From the **Red Hat Certification** web user interface, click the hardware cert on which you want to add the leverage component.
5. Click the **Certification** Section. In the **Progress** tab, go to the **Test Result** column and click on the **Test Result ID** to apply the copied Test Result ID.
6. If the leverage ID is successfully applied you will get a message “**Leverage of System Test successfully applied**”.

CHAPTER 7. REGISTERING SUTS AND PREPARING TO TEST

7.1. OVERVIEW

Our preferred method for adding a certification to the LTS is to use the certification request you created on the Red Hat Certification Catalog as the source of information.

This gives you the benefit of simplified product creation-and-results uploading and enables automatic modification of the local test plan to include only those items required to complete certification.

If for some reason you cannot link to the catalog or you are creating "scratch" test runs not intended for submission, you should use the **Sandbox** option to create local requests unassociated with the online catalog.

Decide the method you wish to use and then follow these steps:

1. If you will be connecting to the Red Hat Certification Catalog, fill in the username and password fields that appear when you go to the LTS in your browser.
They may not appear if you previously canceled the request for login.
2. In that case, click the **Login** link at the top-right of the window and fill in the credentials.
If you will be using the sandbox method for scratch or disconnected testing, click the **Cancel** button if the login screen appears when browsing to the LTS.

7.2. REGISTER A SUT WITH THE LTS

Any system that will undergo testing must be registered with the local test server before testing can begin.

1. Open a web browser and go to the hostname or IP address of the LTS.
If you have not already done so, follow the steps in the [Overview](#) for this section to choose whether or not you will be connected to the catalog, then proceed with these steps:
2. Click on the configuration slider in the upper right-hand corner of the window.
3. Enter the hostname or the IP address of the SUT in the **Register a System** text box that appears.
4. Click **Add** to add the system.
After a brief pause, the SUT will appear under the Registered Systems heading at the top of the page.

Should the command appear to complete without the system appearing in the list of registered systems, click the **refresh** icon in your browser.

Return to the home page of the LTS by clicking the Red Hat Certification graphic at the upper left-hand corner of the page.

7.3. ADD A PRODUCT TO THE LTS FROM THE RED HAT CERTIFICATION CATALOG

1. Open a web browser and go to the hostname or IP address of the LTS.
Follow the steps in the [Overview](#) for this section, if you have not already done so, and provide your username and password to connect to the Catalog, then proceed with these steps:

2. Click the **Certifications** tab on the home page of the LTS if it is not already selected.
3. Click the **New Certification** button.
4. Fill in the Red Hat product information for the product you will be certifying and click **Next**.
5. Use the Vendor, Make, and Name fields to find the certification on the catalog that you wish to test and submit results for, then click the **Next** button.
6. The Certification section of the certification entry should now be highlighted. If it is not, click **Certification** on the left of the page, then click the **Testing** tab.
7. Click the **Add System** button.
 - a. Select the radio button next to the system you want to test.
 - b. Click the **Test** button to proceed with the testing.

7.4. ADD A SANDBOX PRODUCT TO THE LTS

Certifications in the sandbox area do not require network access to the catalog, so it's the ideal method for any "scratch" certifications not intended to be uploaded or for when your test server is unable to reach the catalog due to network security or other reasons.

Follow the steps in the [Overview](#) for this section, if you have not already done so, and cancel the request to log in if it appears, then proceed with these steps:

1. Click the **Sandboxes** tab on the home page of the LTS.
2. Click the **New Sandbox** button.
3. Fill in the Red Hat product information for the product you will be certifying and click **Next**.
4. Choose the Hardware program, give a name to the sandbox entry, and click **Next**.
5. Click the **Add System** button
 - a. Select the radio button next to the system you want to test
 - b. Click the **Test** button to proceed with the sandbox testing.

CHAPTER 8. RUNNING TESTS AND SUBMITTING LOGS FOR REVIEW

8.1. SELECTING AND RUNNING TESTS

There are two types of tests:

- **Automated** tests run when selected without user intervention.
- **Interactive** tests are labeled as such and require additional user input for completion.

Use the following steps to choose which tests to run:

For certifications that are associated with an entry on the catalog

1. Click the **Certifications** tab, then click the name of the Red Hat product on the line that corresponds with your system or component.
2. The **Certification** section of the certification entry should be highlighted when the entry loads. If it is not,
 - a. Click **Certification** on the left of the page.
 - b. Click the **Testing** tab.
 - c. Click the **Continue Testing** button and skip ahead to Step 3, below, to select tests.

For sandbox certifications

1. Click the **Sandboxes** tab, then the name of the Red Hat product on the line that corresponds with your system or component.
2. Click the **Continue Testing** button on the **Testing** tab.
3. Choose the test(s) you wish to run by selecting the check box next to the test(s) and begin the run by clicking the **Run Selected** button.

Note

- You can run tests in any order and in any combination.
- If the test is interactive, you will be prompted for additional information (insert or remove a USB3 device, for example) during the test.
- The tests will run and display their progress on screen.
- After the run finishes, it will appear in the list of runs and the **Continue Testing** button will reappear.
- You can then run additional tests or view the logs from the previous run(s) and submit results.

8.2. VIEW THE TEST LOGS AND SUBMIT THE LOGS FOR REVIEW

You can see the test runs on the **Testing** tab of the certification on your LTS. Click on the entries under **Run**, such as **2017-06-30 12:59:04** to see what tests were performed and whether they passed or failed. Clicking a result will give more detailed information about that run of the test.

To submit results from a run where the certification is associated with the catalog:

1. Click on the run you wish to submit.
Optionally, you can use the drop-down boxes under the **Save Assignment** column to choose which test plan item that test result will satisfy.
2. Click the **Submit Results** button at the bottom of the page to send the results from the displayed run to the Red Hat Certification Catalog.

To submit results from a sandbox run:

1. Click on the run you wish to submit.
2. In the **Action** drop-down box, select **Download** and download the results file.
3. Go to Red Hat Certification and open the certification.
4. Go to the **Testing** tab and select **Upload Results File**.

To submit results using CLI:

To submit the test logs using Red Hat Certification CLI, run the # **rhcert-cli submit** command on the SUT.

Type your Red Hat account credentials previously enabled for certification in the Red Hat Catalog Username and Password. The **Certification ID** is generated when you successfully create a certification request. Type the ID of the certification request in the Certification ID dialog box.

The # **rhcert-cli submit** command works only if the image has a network that can connect to the Red Hat services. The command submits the latest timestamped test logs on your host/image to Red Hat certification services for review. The test log file is reviewed by Red Hat certification operations team. The certification results are displayed on Red Hat Certification web user interface.

If SUT does not have internet access, save the test logs on the SUT using the # **rhcert-cli save -server [hostname/IP address of LTS]** command. The rhcert-cli save command can also be implemented on the LTS.

8.3. RED HAT REVIEW OF TEST RESULTS

After you submit your results, the review team will analyze their contents and award credit for each passing test that is part of the test plan.

As they verify each passing test, the team sets each test plan item to **Confirmed** on the certification site's test plan, which you can see under the **Results** tab on the catalog. This allows you to see at a glance which tests are outstanding and which have been verified as passing.

If any problems are found, the review team will update the certification request with a question, which will automatically be emailed to the person who submitted the cert.

You can see all the discussion, and respond to or ask any questions, on the **Dialog** tab of the certification.

8.4. COMPLETING CERTIFICATIONS

A certification is complete once all the items on the official test plan have been reviewed and found to have passing results. At this point the certification can be closed and published, or closed and left unpublished.

Supplemental certifications always remain unpublished, and system or component certs can also be closed and left unpublished if the vendor does not want to publicly advertise the certification status or the existence of the system/component (most certifications are closed and published).

The system information and the discussions between the tester and review team will not be visible to the general public in a published cert. All that customers can see when viewing published certs is basic information about the system.

APPENDIX A. REFERENCE MATERIAL

A.1. APPENDIXES

A.1.1. Hardware Test Procedures

In this section we give more detailed information about each of the tests for hardware certification. Each test section uses the following format:

What the test covers: This section lists the types of hardware that this particular test is run on.

What the test does: This section explains what the test scripts do. Remember, all the tests are python scripts and can be viewed in the directory `/usr/lib/python2.7/site-packages/rhcert/suites/hwcert/tests` if you want to know exactly what commands we are executing in the tests.

Preparing for the test: This section talks about the steps necessary to prepare for the test. For example, it talks about having a USB device on hand for the USB test and blank discs on hand for rewritable optical drive tests.

Executing the test: This section identifies whether the test is interactive or non-interactive and explains what command is necessary to run the test.

Run Time: This section explains how long a run of this test will take. Timing information for the **info** test is mentioned in each section as it is a required test for every run of the test suite.

A.1.1.1. 1GigEthernet

What the test covers: The **1GigEthernet** test is run on all wired Ethernet connections with a maximum connection speed of 1 gigabit/sec. Connection speed is determined by parsing the "Speed" line in the output of **ethtool**.

What the test does: This test adds link speed detection to the existing **network** test. In addition to passing all the existing network test items, this test must detect a throughput of 1Gb/s (with a margin for overhead) in order to pass. Please see [Section A.1.1.27, "network"](#) for information on the rest of the test functionality.

A.1.1.2. 10GigEthernet

What the test covers: The **10GigEthernet** test is run on all wired Ethernet connections with a maximum connection speed of 10 gigabits/sec. Connection speed is determined by parsing the "Speed" line in the output of **ethtool**.

What the test does: This test adds link speed detection to the existing **network** test. In addition to passing all the existing network test items, this test must detect a throughput of 10Gb/s (with a margin for overhead) in order to pass. Please see [Section A.1.1.27, "network"](#) for information on the rest of the test functionality.

A.1.1.3. 20GigEthernet

What the test covers: The **20GigEthernet** test is run on all wired Ethernet connections with a maximum connection speed of 20 gigabits/sec. Connection speed is determined by parsing the "Speed" line in the output of **ethtool**.

What the test does: This test adds link speed detection to the existing **network** test. In addition to passing all the existing network test items, this test must detect a throughput of 20Gb/s (with a margin for overhead) in order to pass. Please see [Section A.1.1.27, “network”](#) for information on the rest of the test functionality.

A.1.1.4. 25GigEthernet

What the test covers: The **25GigEthernet** test is run on all wired Ethernet connections with a maximum connection speed of 25 gigabits/sec. Connection speed is determined by parsing the "Speed" line in the output of **ethtool**.

What the test does: This test adds link speed detection to the existing **network** test. In addition to passing all the existing network test items, this test must detect a throughput of 25Gb/s (with a margin for overhead) in order to pass. Please see [Section A.1.1.27, “network”](#) for information on the rest of the test functionality.

A.1.1.5. 40GigEthernet

What the test covers: The **40GigEthernet** test is run on all wired Ethernet connections with a maximum connection speed of 40 gigabits/sec. Connection speed is determined by parsing the "Speed" line in the output of **ethtool**.

What the test does: This test adds link speed detection to the existing **network** test. In addition to passing all the existing network test items, this test must detect a throughput of 40Gb/s (with a margin for overhead) in order to pass. Please see [Section A.1.1.27, “network”](#) for information on the rest of the test functionality.

A.1.1.6. 50GigEthernet

What the test covers: The **50GigEthernet** test is run on all wired Ethernet connections with a maximum connection speed of 50 gigabits/sec. Connection speed is determined by parsing the "Speed" line in the output of **ethtool**.

What the test does: This test adds link speed detection to the existing **network** test. In addition to passing all the existing network test items, this test must detect a throughput of 50Gb/s (with a margin for overhead) in order to pass. Please see [Section A.1.1.27, “network”](#) for information on the rest of the test functionality.

A.1.1.7. 100GigEthernet

What the test covers: The **100GigEthernet** test is run on all wired Ethernet connections with a maximum connection speed of 100 gigabits/sec. Connection speed is determined by parsing the "Speed" line in the output of **ethtool**.

What the test does: This test adds link speed detection to the existing **network** test. In addition to passing all the existing network test items, this test must detect a throughput of 100Gb/s (with a margin for overhead) in order to pass. Please see [Section A.1.1.27, “network”](#) for information on the rest of the test functionality.



NOTE

For systems with 50 and 100Gb/s Ethernet options, testing is not required until September 9th 2016. A knowledgebase entry will be added to certifications without passing test results.

A.1.1.8. audio

What the test covers: Removable sound cards and integrated sound devices are tested with the **audio** test. The test is scheduled when the hardware detection routines find the following strings in the udev database:

```
E: SUBSYSTEM=sound
E: SOUND_INITIALIZED=1
```

You can see these strings and the strings that trigger the scheduling of the other tests in this guide in the output of the command **udevadm info --export-db**.

What the test does: The test plays a prerecorded sound (guitar chords or a recorded voice) while simultaneously recording it to a file, then it plays back the recording and asks if you could hear the sound.

Preparing for the test: Before you begin your test run, you should ensure that the audio test is scheduled and that the system can play and record sound. Contact your support contact at Red Hat for further assistance if the test does not appear on a system with installed audio devices. If the test is correctly scheduled, continue on to learn how to manually test the playback and record functions of your sound device.

With built-in speakers present or speakers/headphones plugged into the headphone/line-out jack, playback can be confirmed before testing in these ways:

1. In Red Hat Enterprise Linux 6, right-click on the volume icon at the top of the GUI window and choose **Sound Preferences**. With the tool open, click on the **Hardware** tab, select the sound card you wish to test, and adjust the output volume to an appropriate level. Next, click the **Test Speakers** button. In the window that appears, click the test buttons to generate sounds. Close the test window and exit the sound settings when finished.
2. In Red Hat Enterprise Linux 7, right-click on the volume icon at the top of the GUI window and choose **Sound Settings**. With the tool open, click on the **Output** tab, select the sound card you wish to test, and adjust the output volume to an appropriate level. Next, click the **Test Speakers** button. In the window that appears, click the test buttons to generate sounds. Close the test window and exit the sound settings when finished.

If no sound can be heard, ensure that the speakers are plugged in to the correct port. You can use any line-out or headphone jack (we have no requirement for which port you must use). Make sure sound is not muted and try adjusting the volume on the speakers and in the operating system itself.

If the audio device has record capabilities, these should also be tested before attempting to run the test. Plug a microphone into one of the Line-in or Mic jacks on the system, or you can use the built-in microphone if you are testing a laptop. Again, we don't require you to use a specific input jack; as long as one works, the test will pass.

1. In Red Hat Enterprise Linux 6, right-click on the volume icon at the top of the GUI window and choose **Sound Preferences**. With the tool open, click the **Input** tab, select the appropriate input, and adjust the input volume to 100%. Tap the mic or blow on it, and watch the **Input level** graphic. If you see it moving, the microphone is set up properly. If it does not move, try another input selection and/or microphone port to plug the microphone into.
2. In Red Hat Enterprise Linux 7, right-click on the volume icon at the top of the GUI window and choose **Sound Settings**. With the tool open, click the **Input** tab, select the appropriate input, and adjust the input volume to 100%. Tap the mic or blow on it, and watch the **Input level**

graphic. If you see it moving, the microphone is set up properly. If it does not move, try another input selection and/or microphone port to plug the microphone into.

Contact your support person if you are unable to either hear sound or see the input level display move, as this will lead to a failure of the audio test. If you are able to successfully play sounds and see movement on the input level display when making sounds near the microphone, continue to the next section to learn how to run the test.

Executing the test: The audio test is interactive. Before you execute a test run that includes an audio test, connect the microphone you used for your manual test and place it in front of the speakers, or ensure that the built-in microphone is free of obstructions. Alternatively, you can connect the line-out jack directly to the mic/line-in jack with a patch cable if you are testing in a noisy environment. Check the box next to the test name to indicate it is among the tests to run. Click the button **Run Selected** to continue. The interactive steps are as follows:

1. The system will play sounds and ask if you heard them. Answer *y* or *n* as appropriate. If you decide to use a direct connection between output and input rather than speakers and a microphone, you will need to choose *y* for the answer regardless, as your speakers will be bypassed by the patch cable.
2. The system will next play back the file it recorded. If you heard the sound, answer *y* when prompted. Otherwise, answer *n*.

Run time: The audio test takes less than 1 minute for simultaneous playback and record, then the playback of the recorded sound. The required info test will add about a minute to the overall run time.

A.1.1.9. battery

What the test covers: The **battery** test is only valid for systems that can be powered by a built-in battery use an AC adapter. It does not test external batteries like those found in a UPS, additional internal batteries like the BIOS battery or battery-backed cache, or any other kind of battery that is not providing primary, internal power to the system. The test is scheduled when the hardware detection routines find the following string in the udev database:

```
POWER_SUPPLY_TYPE=Battery
```

What the test does: The test does all its work based on the status of the AC adapter. Testing begins with the AC adapter attached to the system. The test scripts verify the status of the AC adapter and that the battery is present. Then the tester is asked to unplug the adapter, which will cause the battery to begin discharging. The test scripts verify this.

Preparing for the test: The battery test requires that the system be connected via an AC adapter when the test is launched. Ensure that it is connected before proceeding.

Executing the test: The battery test is interactive. Check the box next to the test name to indicate it is among the tests to run. Click the button Run Selected to continue. When the test begins, it will display the current status of the battery (capacity and charging status) and ask for the AC adapter to be unplugged until the battery discharges for 10 mWh. The test will automatically end at that point and the tester should plug the AC adapter back in.

Run time: The time of the battery test varies depending on the discharge and recharge speeds of the battery. It takes about 3 minutes on a 2012-era laptop that emphasizes portability and long battery life over screen size and computing power. Because this test is run on laptops, a **suspend** test must accompany the required info test for each run. The suspend test will add approximately 6 minutes to each test run, and info will add another minute.

A.1.1.10. bluray

What the test covers: All supported optical drives, regardless of formats and features, use the same test methodology, so we are covering all of them in a single section. There are three certification tests for optical media:

- **bluray** - Tests BD-ROM , BD-R and BD-RE media
- **dvd** - Tests DVD-ROM, DVD-R, DVD+R, DVD-RW, and DVD+RW media
- **cdrom** - Tests CD-ROM, CD-R and CD-RW media

Any other disc formats or features like dual-layer (DL) discs, -RAM discs or HD-DVD discs are not tested by the rhcert suite, and can be ignored. The rhcert application determines which of the optical drive tests to schedule, if any, and what type of media to request based on udev information. Here's an example of the udev database on a desktop computer, showing the supported media of the system's CD-RW, DVD+/-RW, BD-RE drive:

```
E: ID_CDROM=1
E: ID_CDROM_CD=1
E: ID_CDROM_CD_R=1
E: ID_CDROM_CD_RW=1
E: ID_CDROM_DVD=1
E: ID_CDROM_DVD_R=1
E: ID_CDROM_DVD_RW=1
E: ID_CDROM_DVD_RAM=1
E: ID_CDROM_DVD_PLUS_R=1
E: ID_CDROM_DVD_PLUS_RW=1
E: ID_CDROM_DVD_PLUS_R_DL=1
E: ID_CDROM_BD=1
E: ID_CDROM_BD_R=1
E: ID_CDROM_BD_RE=1
```

The scripts look for *ID_CDROM=1* before scheduling any of the three optical media tests. If it finds this value, it analyzes the properties to determine which of the three tests to schedule. You can see the drive's *ID_CDROM* properties in the udev output above. These tell the rhcert application that the drive is capable of writing to many different disc formats including CD, DVD and Blu-Ray (BD). From that information we know that the bluray, cdrom and dvd tests will be scheduled, and the test harness decides which feature of the format to test. The following tables explain how the rhcert application makes that determination:

The test suite always attempts to schedule the most advanced media tests first in accordance with the rules in the Policy Guide, which requires testing read, write and erase functionality when all are present. Discs that support rewrite functions include:

- BD-RE (tested as part of the 'bluray' test)
- Either DVD-RW or DVD+RW (tested as part of the 'dvd' test)
- CD-RW (tested as part of the 'cdrom' test)

Only formats supported by the drive are scheduled for testing. If your drive(s) support DVD-RW and DVD+RW, you can use either format of disc during the test. You do not have to test both.

If the drive is not capable of rewrite operations but it does have write-once capabilities for a disc format, the test suite schedules a write-once media test. Discs that support write-once functionality include:

- BD-R (tested as part of the 'bluray' test)
- Either DVD-R or DVD+R (tested as part of the 'dvd' test)
- CD-R (tested as part of the 'cdrom' test)

Only formats supported by the drive are scheduled for testing. If your drive(s) support DVD-R and DVD+R, you can use either format of disc during the test. You do not have to test both.

If the drive is not capable of rewrite or write-once operations but it does have read-only support for a disc format, the test suite schedules a read-only media test. Discs that are read-only include:

- BD-ROM (tested as part of the 'bluray' test)
- DVD-ROM (tested as part of the 'dvd' test)
- CD-ROM (tested as part of the 'cdrom' test)

Only formats supported by the drive are scheduled for testing.

Using the udev data from our example laptop BD/DVD/CD drive from above, we can use this list of discs and tests to determine what types of media are needed. The drive supports all types of Blu-Ray media and since rewritable discs take precedence over write-once or read-only, a BD-RE disc will be needed for the bluray test.

The policy guide was updated at the launch of Red Hat Enterprise Linux 6.3 to reduce the number of optical drive tests that must be performed. Now each controller will only need one test instead of multiple tests of different disc formats. For the example drive shown above, you would run a Blu-Ray rewritable disc test and nothing else. The other tests (CDROM and DVD) are still planned by the rhcert tool, but you do not have to run them. How do you know what drive and which disc type to test? Here is a handy table that explains how it works:

Table A.1. Blank Table of Optical Drive Features

	Rewrite or Write	Read Only
	BD-RE	DVD+/-RW
CD-RW	BD-R	DVD+/-R
CD-R	BD-ROM	DVD-ROM
CD-ROM	Drive 1	
		Drive 2

	Rewrite or Write	Read Only
Drive 3		
	...	
		Drive X

Fill out the table with all the drives you have available to you on your controller. Place an "X" in the column that corresponds with the disc format that each drive supports. When you have finished, choose the drive that has an "X" in the column furthest to the left for your certification testing and be prepared to test that kind of media in the drive. If two or more drives have an "X" in the same leftmost column, you can use either drive for your tests.

Here's an example.

- Drive 1 - A Blu-Ray drive that supports rewriting
- Drive 2 - A CD-ROM drive that supports rewriting
- Drive 3 - A DVD drive that supports rewriting
- Drive 4 - A CD-ROM drive that supports read functions only
- Drive 5 - A Blu-Ray drive that supports writing, but not rewriting

Table A.2. Sample Table of Optical Drive Features

	Rewrite or Write	Read Only
	BD-RE	DVD+/-RW
CD-RW	BD-R	DVD+/-R

	Rewrite or Write	Read Only
CD-R	BD-ROM	DVD-ROM
CD-ROM	Drive 1	X
X	X	X
X	X	X
X	X	Drive 2
	X	X
		X
Drive 3		
X	X	X
X		X
X	Drive 4	
	X	Drive 5
	X	X
X	X	X
X	X	X

For the series of drives in the example chart above, you would choose to do your test with Drive 1, and you would only need to run the bluray test with a BD-RE disc. This is because Drive 1 is the drive with an "X" in the furthest column to the left, and that column corresponds with BD-RE media. No other testing would be required.

What the test does: For read-only drives, it reads data from the disc and copies it to the hard drive. The original data on the disc is then compared to the copy on the hard drive. If all file checksums match, the test passes. Writable media adds a write procedure to the test. A blank writable disc is inserted in the system and data is written to it from the hard drive. The data on the disc is then compared to the data on the hard drive. If the file checksums match, the test passes. Rewritable media adds a disc blank to the

procedure, followed by a write of data from the hard drive and a comparison of the written data to the original. If the blank is successful and the checksums of the newly written files on the disc match those on the hard drive, the test passes. The test also includes disc ejects between each phase (blank, write, compare). The tester will need to insert the disc back into the drive if the drive is not capable of closing the tray by itself, or if it is a slot loading drive.

Executing the test: The bluray test is interactive. Install the proper drive as determined by the table you created. Check the box next to the test name to indicate it is among the tests to run. Click the button **Run Selected** to continue. Follow the directions on screen and choose the proper disc format when prompted (the one corresponding with the leftmost column in the table that has an "X" in it), then insert the correct disc when asked. As the test enters the various phases (blank, write, compare, where applicable), the on-screen display will explain what is happening.

Run time: The run time for all optical drive testing is dependent on the speed of the media and drive. For a 4x DVD-RW disc, the DVD test takes about 10 minutes to write and verify ~1.7GB of data.

A.1.1.11. cdrom

CD drives of all kinds are tested using the same procedures as Blu-Ray drives. Please see [Section A.1.1.10, "bluray"](#) for more information.

A.1.1.12. core

What the test covers: The **core** test examines the system's CPUs and ensures that they are capable of functioning properly under load.

What the test does: The **core** test is actually composed of two separate routines. The first test is designed to detect clock jitter. Jitter is a condition that occurs when the system clocks are out of sync with each other. The system clocks are not the same as the CPU clock speed, which is just another way to refer to the speed at which the CPUs are operating. The jitter test uses the **gettimeofday()** function to obtain the time as observed by each logical CPU and then analyzes the returned values. If all the CPU clocks are within .2 nanoseconds of each other, the test passes. The tolerances for the jitter test are very tight. In order to get good results it's important that the rhcert tests are the only loads running on a system at the time the test is executed. Any other compute loads that are present could interfere with the timing and cause the test to fail. The jitter test also checks to see which clock source the kernel is using. It will print a warning in the logs if an Intel processor is not using TSC, but this will not affect the PASS/FAIL status of the test.

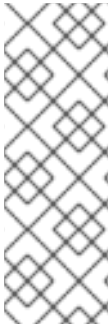
The second routine run in the core test is a CPU load test. It's the test provided by the required **stress** package. The stress program, which is available for use outside the rhcert suite if you are looking for a way to stress test a system, launches several simultaneous activities on the system and then monitors for any failures. Specifically it instructs each logical CPU to calculate square roots, it puts the system under memory pressure by using **malloc()** and **free()** routines to reserve and free memory respectively, and it forces writes to disk by calling **sync()**. These activities continue for 10 minutes, and if no failures occur within that time period, the test passes. Please see the **stress** manpage if you are interested in using it outside of hardware certification testing.

Preparing for the test: The only preparation for the core test is to install a CPU that meets the requirements that are stated in the Policy Guide.

Executing the test: The core test is non-interactive. Check the checkbox next to the test and click the **Run Selected** button to perform the test.

Run time, bare-metal: The core test itself takes about 12 minutes to run on a bare-metal system. The jitter portion of the test takes a minute or two and the stress portion runs for exactly 10 minutes. The required info test will add about a minute to the overall run time.

Run time, full-virt guest: The `fv_core` test takes slightly longer than the bare-metal version, about 14 minutes, to run in a KVM guest. The added time is due to guest startup/shutdown activities and the required info test that runs in the guest. The required info test on the bare-metal system will add about a minute to the overall run time.



NOTE

A note about FV testing times: The first time you run any full-virt test, the test tool will need to acquire the FV guest files. If these files are located on the local test server and you are using 1GbE or faster networking, that will take only a minute or two to transfer the ~300MB of guest files. If the files are retrieved from the Red Hat FTP server, which happens automatically if the guest files are not installed and not found on the local test server, the first runtime will depend on the speed of the FTP transfer. Once the guest files are available on the SUT they will be used for all subsequent runs of `fv_*` tests.

A.1.1.13. cpuscaling

What the test covers: The **cpuscaling** test examines a CPU's ability to increase and decrease its clock speed according to the compute demands placed on it.

What the test does: The test exercises the CPUs at varying frequencies using different scaling governors (the set of instructions that tell the CPU when to change to higher or lower clock speeds and how fast to do so) and measures the difference in the time that it takes to complete a standardized workload. The test is scheduled when the hardware detection routines find the following directories in `/sys` containing more than one cpu frequency:

```
/sys/devices/system/cpu/cpuX/cpufreq
```

The cpuscaling test is planned once per package, rather than being listed once per logical CPU. When the test is run, it will determine topology via `/sys/devices/system/cpu/cpuX/topology/physical_package_id`, and run the test in parallel for all the logical CPUs in a particular package.

The test procedure for each CPU package is as follows:

The test uses the values found in the `sysfs` filesystem to determine the maximum and minimum CPU frequencies. You can see these values for any system with this command:

```
# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
```

There will always be at least two frequencies displayed here, a maximum and a minimum, but some processors are capable of finer CPU speed control and will show more than two values in the file. Any additional CPU speeds between the max and min are not specifically used during the test, though they may be used as the CPU transitions between max and min frequencies. The test procedure is as follows:

1. The test records the maximum and minimum processor speeds from the file `/sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies`.
2. The userspace governor is selected and maximum frequency is chosen.
3. Maximum speed is confirmed by reading all processors' `/sys/devices/system/cpu/cpuX/cpufreq/scaling_cur_freq` value. If this value does not match the selected frequency, the test will report a failure.

4. Every processor in the package is given the simultaneous task of calculating pi to 2×10^{12} digits. The value for the pi calculation was chosen because it takes a meaningful amount of time to complete (about 30 seconds).
5. The amount of time it took to calculate pi is recorded for each CPU, and an average is calculated for the package.
6. The userspace governor is selected and the minimum speed is set.
7. Minimum speed is confirmed by sysfs data, with a failure occurring if any CPU is not at the requested speed.
8. The same pi calculation is performed by every processor in the package and the results recorded.
9. The ondemand governor is chosen, which throttles the CPU between minimum and maximum speeds depending on workload.
10. Minimum speed is confirmed by sysfs data, with a failure occurring if any CPU is not at the requested speed.
11. The same pi calculation is performed by every processor in the package and the results recorded.
12. The performance governor is chosen, which forces the CPU to maximum speed at all times.
13. Maximum speed is confirmed by sysfs data, with a failure occurring if any CPU is not at the requested speed.
14. The same pi calculation is performed by every processor processor and the results recorded.

Now the analysis is performed on the three subsections. In steps one through eight we obtain the pi calculation times at maximum and minimum CPU speeds. The difference in the time it takes to calculate pi at the two speeds should be proportional to the difference in CPU speed. For example, if a hypothetical test system had a max frequency of 2GHz and a min of 1GHz and it took the system 30 seconds to run the pi calculation at max speed, we would expect the system to take 60 seconds at min speed to calculate pi. We know that for various reasons perfect results will not be obtained, so we allow for a 10% margin of error (faster or slower than expected) on the results. In our hypothetical example, this means that the minimum speed run could take between 54 and 66 seconds and still be considered a passing test (90% of 60 = 54 and 110% of 60 = 66).

In steps nine through eleven, we test the pi calculation time using the ondemand governor. This confirms that the system can quickly increase the CPU speed to the maximum when work is being done. We take the calculation time obtained in step eleven and compare it to the maximum speed calculation time we obtained back in step five. A passing test has those two values differing by no more than 10%.

In steps twelve through fourteen, we test the pi calculation using the performance governor. This confirms that the system can hold the CPU at maximum frequency at all times. We take the pi calculation time obtained in step 14 and compare it to the maximum speed calculation time we obtained back in step five. Again, a passing test has those two values differing by no more than 10%.

An additional portion of the cpuscaling test runs when an Intel processor with the TurboBoost feature is detected by the presence of the **ida** CPU flag in **/proc/cpuinfo**. This test chooses one of the CPUs in each package, omitting CPU0 for housekeeping purposes, and measures the performance using the ondemand governor at maximum speed. It expects a result of at least 5% faster performance than the previous test, when all the cores in the package were being tested in parallel.

Preparing for the test: To prepare for the test, ensure that CPU frequency scaling is enabled in the BIOS and ensure that a CPU is installed that meets the requirements explained in the Policy Guide.

Executing the test: The cpuscaling test is non-interactive. Check the checkbox next to the test and click the **Run Selected** button to perform the test.

Run time: The cpuscaling test takes about 42 minutes for a 2013-era, single CPU, 6-core/12-thread 3.3GHz Intel-based workstation running Red Hat Enterprise Linux 6.4, AMD64 and Intel 64. Systems with higher core counts and more populated sockets will take longer. The required info test will add about a minute to the overall run time.

A.1.1.14. dvd

DVD drives of all kinds are tested using the same procedures as Blu-Ray drives. Please see [Section A.1.1.10, “bluray”](#) for more information.

A.1.1.15. Ethernet

What the test covers: The **Ethernet** test only appears when the speed of a network device is not recognized by the test suite. This may be due to an unplugged cable or some other fault is preventing the proper detection of the connection speed. Please exit the test suite, check your connection, and run the test suite again when the device is properly connected. If the problem persists, contact your Red Hat support representative for assistance.

The example below shows a system with two gigabit Ethernet devices, eth0 and eth1. Device eth0 is properly connected, but eth1 is not plugged in.

The output of the **ethtool** command shows the expected gigabit Ethernet speed of 1000Mb/s for eth0:

```
# ethtool eth0
Settings for eth0:
  Supported ports: [ TP ]
  Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full
  Supported pause frame use: No
  Supports auto-negotiation: Yes
  Advertised link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full
  Advertised pause frame use: No
  Advertised auto-negotiation: Yes
  Speed: 1000Mb/s
  Duplex: Full
  Port: Twisted Pair
  PHYAD: 2
  Transceiver: internal
  Auto-negotiation: on
  MDI-X: on
  Supports Wake-on: pumbg
  Wake-on: g
  Current message level: 0x00000007 (7)
                        drv probe link
  Link detected: yes
```

But on eth1 the **ethtool** command shows an unknown speed, which would cause the **Ethernet** test to be planned.

```
# ethtool eth1
Settings for eth1:
  Supported ports: [ TP ]
  Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full

  Supported pause frame use: No
  Supports auto-negotiation: Yes
  Advertised link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full

  Advertised pause frame use: No
  Advertised auto-negotiation: Yes
  Speed: Unknown!
  Duplex: Unknown! (255)
  Port: Twisted Pair
  PHYAD: 1
  Transceiver: internal
  Auto-negotiation: on
  MDI-X: Unknown
  Supports Wake-on: pumbg
  Wake-on: g
  Current message level: 0x00000007 (7)
                        drv probe link
  Link detected: no
```

A.1.1.16. expresscard

What the test covers: The **expresscard** test looks for devices with both types of ExpressCard interfaces, USB and PCI Express (PCIe), and confirms that the system can communicate through both. ExpressCard slot detection is not as straightforward as detecting other devices in the system. ExpressCard was specifically designed to not require any kind of dedicated bridge device. It's merely a novel form factor interface that combines PCIe and USB. Because of this, there is no specific "ExpressCard slot" entry that we can see in the output of udev. We decided to schedule the test on systems that contain a battery, USB and PCIe interfaces, as we have seen no devices other than ExpressCard-containing laptops with this combination of hardware.

What the test does: The test first takes a snapshot of all the devices on the USB and PCIe buses using the **lsusb** and **lspci** commands. It then asks the tester how many ExpressCard slots are present in the system. The tester is asked to insert a card in one of the slots. The system scans the USB and PCIe buses and compares the results to the original **lsusb** and **lspci** output to detect any new devices. If a USB device is detected, the system asks you to remove the card and insert a card with a PCIe interface into the same slot. If a PCIe-based card is detected, the system asks you to remove it and insert a USB-based card into the same slot. If a card is inserted with both interfaces (a docking station card, for example), it fulfills both testing requirements for the slot at once. This procedure is repeated for all slots in the system.

Preparing for the test: You will need ExpressCard cards with USB and PCIe buses. This can be two separate cards or one card with both interfaces. Remove all ExpressCard cards before running the test.

Executing the test: The **expresscard** test is interactive. Check the box next to the test name to indicate it is among the tests to run. Click the button Run Selected to continue. It will prompt you to remove all

ExpressCards, then ask for permission to load the PCI Express hotplug module (pciehp) if it is not loaded. PCIe hotplug capabilities are needed in order to add or remove PCIe-based ExpressCard cards while the system is running. Next the test will ask you for the number of ExpressCard slots in the system, followed by prompts to insert and remove cards with both types of interfaces (USB and PCIe) in any order.

A.1.1.17. fv_core

The **fv_core** test is a wrapper that launches the FV guest and runs a **core** test on it. Please see [Section A.1.1.12, “core”](#) for information on the test methodology and run times.

A.1.1.18. fv_memory

The **fv_memory** test is a wrapper that launches the FV guest and runs a **memory** test on it. Please see [Section A.1.1.26, “memory”](#) for information on the test methodology and run times.

A.1.1.19. fv_network (Optional for SR-IOV)

The **fv_network** test is a wrapper that launches the FV guest and runs a **network** test on it. It is useful for verifying the function of one or more network devices that support SR-IOV.

What the test covers: The test covers virtual function network devices on SR-IOV capable systems. Systems without SR-IOV may run the test too, but it will only verify the function of the standard virtual network hardware.

What the test does: Please see [Section A.1.1.27, “network”](#) for information on the test methodology and run times.

Preparing for the test: Assign a virtual function (VF) from a NIC to the guest. Directions on how to configure VFs can be found in the [Using SR-IOV section](#) of the [Virtualization Deployment and Administration Guide](#).

Executing the test: The **fv_network** test is non-interactive. After properly assigning a VF to the guest, check the checkbox next to the test and click the **Run Selected** button to perform the test.

A.1.1.20. fv_storage (Optional)

The **fv_storage** test is a wrapper that launches the FV guest and runs a **storage** test on it. It is not required for certification at this time.

A.1.1.21. infiniband connection

What the test does: The Infiniband Connection test runs the following subtests to ensure a baseline functionality using, when appropriate, the ip address selected from the dropdown at the onset of the test:

1. Ping test
Runs ping from the starting IP address of the device being tested on the SUT to the selected IP address of the LTS.
2. Rping test
Runs rping on LTS and SUT using the selected LTS IP address, then compares results to verify it ran to completion.
3. Rcopy test

Runs rcopy on LTS and SUT, sending a randomly generated file and comparing md5sums on LTS and SUT to verify successful transfer.

4. Rdma-ndd service test

Verifies stop, start and restart service commands function as expected.

5. Opensm service test

Verifies stop, start and restart service commands function as expected.

6. LID verification test

Verifies that the LID for the device is set and not the default value.

7. Smpquery test

Runs smpquery on LTS using device and port for another verification the device/port has been registered with the fabric.

Preparing for the test: Ensure that the LTS and SUT are separate machines, on the same fabric(s).

Executing the test: This is an interactive test. Check the box next to the test name to indicate it is among the tests to run. Click the button Run Selected to continue. You will be prompted with a dropdown to select an ip address (an ip address of LTS) in which to perform the tests using. Select an ip address corresponding to a device on the same fabric of the SUT device you are running the test for.

Manually adding and running the test: Use the following command to manually add the infinibandconnectiontest:

```
rhcert-cli plan --add --test infinibandconnectiontest --device <device name>_devicePort_<port number>
```

Use the following command to manually run the infinibandconnectiontest:

```
rhcert-cli run --test InfinibandConnectionTest --server <LTS IP addr>
```

Run time: This test takes less than 10 minutes to run.

Reference

See [Understanding InfiniBand and RDMA technologies](#) for more information.

A.1.1.22. info

What the test does: The **info** test is a part of all results packages. It's run automatically along with any other test that is being performed and is a required part of every results package. If you attempt to submit a package that contains no info test, the package will be rejected. The test performs several different tasks. If any of these tasks fail, the info test fails:

1. Confirm that **/proc/sys/kernel/tainted** is zero, indicating a non-tainted kernel.
2. Confirm that package verification with **rpm -V** shows that no files have been modified.
3. Confirm that **rpm -qa kernel** shows that the buildhost of the kernel package is a redhat.com machine.
4. Record the boot parameters from **/proc/cmdline** for later analysis by our review team.
5. Confirm that **rpm -V redhat-certification** shows that no modifications have been made to any of the certification test suite files.

6. Confirm that all the modules shown by **lsmod** show up in a listing of the kernel files with the command **rpm -ql kernel**.
7. Confirm that all modules are on the kABI whitelist.
8. Confirm that the module vendor and buildhost are appropriate Red Hat entries.
9. Confirm that the kernel is the GA kernel of the Red Hat minor release. The verification is attempted with data from the **redhat-certification-information** package. Internet verification (direct routing/dns resolution have to work, or environment variable 'ftp_proxy=http://proxy.domain:80' has to be set) is attempted if the kernel is not present in the **redhat-certification-information** package.

After performing those tasks, the system gathers a sosreport and the output of **dmidecode**. These are used by our review team to help them in their analysis of the test results.

Run time: The info test takes around 1 minute on a 2013-era, single CPU, 3.3GHz, 6-core/12-thread Intel workstation with 8 GB of RAM running Red Hat Enterprise Linux 6.4, AMD64 and Intel 64 that was installed using the kickstart files in this guide. The time will vary depending on the speed of the machine and the quantity of RPM files that are installed.

A.1.1.23. iwarp connection

What the test does: The IWarp Connection test runs the following subtests to ensure a baseline functionality using, when appropriate, the IP address selected from the dropdown at the onset of the test:

1. Ping test - Runs ping from the starting IP address of the device being tested on the SUT to the selected IP address of the LTS.
2. Rping test - Runs rping on LTS and SUT using the selected LTS IP address, then compares results to verify it ran to completion.
3. Rcopy test - Runs rcopy on LTS and SUT, sending a randomly generated file and comparing md5sums on LTS and SUT to verify successful transfer.
4. Ethtool test - Runs the ethtool command passing in the detected net device of the roce device.

Preparing for the test: Ensure that the LTS and SUT are separate machines, on the same fabric(s).

Executing the test: This is an interactive test. Check the box next to the test name to indicate it is among the tests to run. Click the button Run Selected to continue. You will be prompted with a dropdown to select an ip address (an ip address of LTS) in which to perform the tests using. Select an ip address corresponding to a device on the same fabric of the SUT device you are running the test for.

Manually adding and running the test: Use the following command to manually add the iwarpconnectiontest:

```
rhcert-cli plan --add --test iWarpConnectionTest --device <device name>_devicePort_<port number>_netDevice_<net device here>
```

Use the following command to manually run the iwarpconnectiontest:

```
rhcert-cli run --test iWarpConnectionTest --server <LTS IP addr>
```

Run time: This test takes less than 10 minutes to run.

Reference

See [Understanding InfiniBand and RDMA technologies](#) for more information.

A.1.1.24. kdump

What the test covers: The **kdump** test verifies the ability of a system to capture a vmcore after a crash using the **kdump** utility. There are two entries in the local test plan, one for local core file storage and one for the remote copying of a vmcore via NFS to the test server.

What the test does: The test will crash the system and write a vmcore to **/var/crash**. It will crash the system a second time and write a vmcore to the **/var/www/hwcert/export** directory on the network / kdump server system. After each of the two actions occurs, the test server program will confirm that the system only did the things it was scheduled to do, e.g. it checks that only two reboots occurred when each panic was triggered.

Preparing for the test: Ensure that the system is connected to the network before running the test. All parameters will be automatically set by the test server.

Executing the test: The kdump test is interactive. Check the box next to the test name to indicate it is among the tests to run. Click the button **Run Selected** to continue. The system will ask you to click the **Yes** or **No** button to trigger the crash when the kdump test is run. The discs will sync and the **vmcore** file will be saved. You will see a series of messages including "Waiting for response", "Waiting for connection", and finally, "ready" as the test server waits for completion of the task. After the core is saved, the system under test will reboot and the rhcert application will be ready for the next test. The rhcert server will verify the **vmcore** file is present and valid. It will then repeat the crash, this time exporting the **vmcore** file to the test server, when you click the button next to the NFS version of tes test.

Run time: The kdump test run time is highly variable. It is dependent on the amount of RAM in the SUT, the speed of the disks in both the SUT and the test server, the speed of the network connection to the test server, and the time it takes to reboot the SUT. For a 2013-era workstation with 8GB of RAM, a 7200 RPM 6Gb/s SATA drive, a gigabit Ethernet connection to the test server and a 1.5 minute reboot time, a local kdump test can complete in about 4 minutes, including the reboot. The same 2013-era workstation can complete a NFS kdump test in about 5 minutes to a similarly equipped network test server. The required info test will add about a minute to the overall run time.

A.1.1.25. lid

What the test covers: The **lid** test is only valid for systems that have integrated displays and therefore have a lid that can be opened and closed. The lid is detected by searching the udev database for a device with "lid" in its name:

```
E: NAME="Lid Switch"
```

What the test does: The test ensures that the system can determine when its lid is closed and when it is open via parameters in udev, and that it can turn off the display's backlight when the lid is closed.

Preparing for the test: To prepare for the test, ensure that the power management settings do not put the system to sleep or into hibernation when the lid is closed. In Red Hat Enterprise Linux 6, right-click on the battery icon in the panel and choose Preferences. On the **AC Power** tab, select *Blank screen* as the action that occurs when the lid is closed. In Red Hat Enterprise Linux 7, use the **Tweak Tool** to disable suspend or hibernate on lid close. Make sure the lid is open before you start the test run.

Executing the test: The lid test is interactive. Check the box next to the test name to indicate it is among the tests to run. Click the button **Run Selected** to continue. You will be asked if you are ready to begin the test, so answer *Yes* to continue. Close the lid when prompted, watching to see if the backlight

turns off. You may have to look through the small space between the keyboard and lid when the laptop is closed to verify that the backlight has turned off. Answer **Yes** if the backlight turns off or **No** if the backlight does not turn off.

Run time: The lid test takes about 30 seconds to perform, essentially the time it takes to close the lid just enough to have the backlight turn off. Because this test is run on laptops, a suspend test must accompany the required info test for each run. The suspend test will add approximately 6 minutes to each test run, and info will add another minute.

A.1.1.26. memory

What the memory test covers: The **memory** test is used to test system RAM. It does not test USB flash memory, SSD storage devices or any other type of RAM-based hardware, it only tests main memory.

What the test does: The test uses the file `/proc/meminfo` to determine how much memory is installed in the system. Once it knows how much is installed, it checks to see if the system architecture is 32-bit or 64-bit. Then it determines if swap space is available or if there is no swap partition. The test runs either once or twice with slightly different settings depending on whether or not the system has a swap file:

1. If swap is available, allocate more RAM to the memory test than is actually installed in the system. This forces the use of swap space during the run.
2. Regardless of swap presence, allocate as much RAM as possible to the memory test while staying below the limit that would force out of memory (OOM) kills. This version of the test always runs.

In both iterations of the memory test, **malloc()** is used to allocate RAM, the RAM is dirtied with a write of an arbitrary hex string (0xDEADBEEF), and a test is performed to ensure that 0xDEADBEEF is actually stored in RAM at the expected addresses. The test calls **free()** to release RAM when testing is complete. Multiple threads or multiple processes will be used to allocate the RAM depending on whether the process size is greater than or less than the amount of memory to be tested.

Preparing for the test: Install the correct amount of RAM in the system in accordance with the rules in the Policy Guide.

Executing the test: The memory test is non-interactive. Check the checkbox next to the test and click the **Run Selected** button to perform the test.

Run time, bare-metal: The memory test takes about 16 minutes to run on a 2013-era, single CPU, 6-core/12-thread 3.3GHz Intel-based workstation with 8GB of RAM running Red Hat Enterprise Linux 6.4, AMD64 and Intel 64. The test will take longer on systems with more RAM. The required info test will add about a minute to the overall run time.

Run time, full-virt guest: The `fv_memory` test takes slightly longer than the bare-metal version, about 18 minutes, to run in a guest. The added time is due to guest startup/shutdown activities and the required info test that runs in the guest. The required info test on the bare-metal system will add about a minute to the overall run time. The `fv_memory` test run times will not vary as widely from machine to machine as the bare-metal memory tests, as the amount of RAM assigned to our pre-built guest is always the same. There will be variations caused by the speed of the underlying real system, but the amount of RAM in use during the test won't change from machine to machine.

**NOTE**

A note about FV testing times: The first time you run any full-virt test, the system under test will need to acquire the FV guest files. If these files are located on the local test server and you are using 1GbE or faster networking, that will take only a minute or two to transfer the ~300MB of guest files. If the files are retrieved from the Red Hat FTP server, which happens automatically if the guest files are not installed or not found on the local test server, the first runtime will depend on the speed of the FTP transfer. Once the guest files are installed, they will be used for all subsequent runs of `fv_*` tests.

A.1.1.27. network

What the test covers: The **network** test is used to test devices whose function is transferring data over a network. This includes wired Ethernet cards, wireless Ethernet cards, virtual network devices on systems that support SR-IOV, and InfiniBand cards if IB is being used as a network protocol. The test will appear as **network** for non-Ethernet devices, or as different names for Ethernet or Wi-Fi devices:

**NOTE**

If a device's PCI class code is 60A00 or C0600 or if the device driver is split into modules such as `mlx4_core`, `mlx5_core` or `mlx5_ib`, the suite will plan Infiniband tests.

- **1GigEthernet** - The network test with added speed detection for 1 gigabit Ethernet connections.
- **10GigEthernet** - The network test with added speed detection for 10 gigabit Ethernet connections.
- **20GigEthernet** - The network test with added speed detection for 20 gigabit Ethernet connections.
- **25GigEthernet** - The network test with added speed detection for 25 gigabit Ethernet connections.
- **40GigEthernet** - The network test with added speed detection for 40 gigabit Ethernet connections.
- **50GigEthernet** - The network test with added speed detection for 50 gigabit Ethernet connections.
- **100GigEthernet** - The network test with added speed detection for 100 gigabit Ethernet connections.

**NOTE**

For systems with 50 and 100Gb/s Ethernet options, testing is not required until September 9th 2016. A knowledgebase entry will be added to certifications without passing test results.

**NOTE**

If you see a test named **Ethernet** in your local test plan, that is an indication that the test suite did not recognize the speed for that device. Please check the connection before attempting to test that particular device. See [Section A.1.1.15, “Ethernet”](#) for more information.

- **WirelessG** - The network test with added speed detection for 802.11g wireless Ethernet connections.
- **WirelessN** - The network test with added speed detection for 802.11n wireless Ethernet connections.
- **WirelessAC** - The network test with added speed detection for 802.11ac wireless Ethernet connections.

What the test does: The test gathers information on all the network devices and runs this procedure:

1. Bounce the interface (**ifdown**, **ifup**) being tested, as long as the root partition is not on an NFS mount. If we were running on NFS root, the system would never come back after losing its connection to root.
2. **ifdown** all interfaces not under test.
3. Create a test file of random data (using **/dev/urandom**) the size of which is tuned to the speed of your NIC.
4. TCP testing - A TCP latency test (**lat_tcp**) is run 5 times. This test watches to see if the system runs into any OS timeouts, which would cause the test to fail. It's followed by a TCP bandwidth test (**bw_tcp**). For wired devices, we expect the speed to be close to the theoretical maximum.
5. UDP testing - A UDP latency test (**lat_udp**) is run and the script watches to see if the system runs into any OS timeouts.
6. HTTP file transfer testing - The script uploads the random testfile created in step three via HTTP multi-part form enclosure, then downloads it via HTTP GET. It times how long it takes to upload and download the file, and verifies the contents of the original to the second generation copy.
7. ICMP (ping) test - The script causes a ping flood at the default packet size to make sure nothing in the system fails (the system should not restart/reset or oops or anything else that indicates the inability to withstand a ping flood.). 5000 packets are sent, and a 100% success rate is expected. The test will retry 5 times for an acceptable success rate.
8. The final action of the test is to bring all interfaces back to where they started, either active or inactive depending on their state when the test was launched.

Preparing for testing wired devices: You may test as many network devices from the official test plan as you wish in each run of the test suite. Connect each device at its native (maximum) speed or the test will fail. Ensure that the hwcert network test server is up and running before beginning, and make sure that each network device has an IP address assigned either statically or via DHCP.

If any network devices support partitioning, we need to see them demonstrate both full-speed data transfer and the partitioning function in one or more runs of the network test. This requirement will be accounted for in the official test plan by having two entries for each NIC that supports partitioning. If the NIC can run at full speed while it's partitioned, please configure a partition with the NIC running at its native speed and perform your network tests in that configuration. This single test run will satisfy both official test plan entries for the NIC.

If the NIC cannot run at full speed while it's partitioned, please perform one network test without partitioning so that we can see full-speed operation, and then perform another network test with partitioning enabled so that we can see a demonstration of the partitioning function. We recommend that you choose either 1Gb/s or 10Gb/s for your partitioned configuration so that it conforms to one of our existing network speed tests.

Preparing for testing wireless Ethernet devices: In Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7, any system with a supported wireless card will automatically receive any necessary firmware package(s) at install time and all configuration of the cards can be done with the NetworkManager graphical tool. Simply select an SSID on a test network that does not require any additional user input during up/down operations (no authentication requests, VPN login, etc.) and you can run the test as explained in the "Executing the test" section below.

**NOTE**

Based on the wireless card which is being tested, the wireless access point that you connect to should be capable of performing WirelessG, WirelessN and WirelessAC network tests.

Executing the test: The network test is non-interactive. Check the checkbox next to the test and click the **Run Selected** button to perform the test.

Run time: The network test takes about 21 minutes for each PCIe-based, gigabit, wired Ethernet card that is being tested. We'll add 10GbE test times and wireless times at a future date. The required info test will add about a minute to the overall run time.

A.1.1.28. omnipath connection

What the test does: The **Omnipath Connection** test runs the following subtests to ensure a baseline functionality using, when appropriate, the ip address selected from the dropdown at the onset of the test:

1. Ping test - Runs ping from the starting IP address of the device being tested on the SUT to the selected IP address of the LTS.
2. Rping test - Runs rping on LTS and SUT using the selected LTS IP address, then compares results to verify it ran to completion.
3. Rcopy test - Runs rcopy on LTS and SUT, sending a randomly generated file and comparing md5sums on LTS and SUT to verify successful transfer.
4. Rdma-ndd service test - Verifies stop, start and restart service commands function as expected.
5. Opensm service test - Verifies stop, start and restart service commands function as expected.
6. LID verification test - Verifies that the LID for the device is set and not the default value.
7. Link speed test - Verifies that the detected link speed is 100Gb.
8. Smpquery test - Runs smpquery on LTS using device and port for another verification the device/port has been registered with the fabric.

Preparing for the test: Ensure that the LTS and SUT are separate machines, on the same fabric. You need to install **opa-basic-tools** on the LTS from the [Downloads](#) section of Red Hat customer portal web page.

Executing the test: This is an interactive test. Check the box next to the test name to indicate it is among the tests to run. Click the button Run Selected to continue. You will be prompted with a dropdown to select an ip address (an ip address of LTS) in which to perform the tests using. Select an ip address corresponding to a device on the same fabric of the SUT device you are running the test for.

Manually adding and running the test: Use the following command to manually add the OmnipathConnectionTest:

```
rhcert-cli plan --add --test ominpathconnectiontest --device <device
name>_devicePort_<port number>
```

Use the following command to manually run the OmnipathConnectionTest:

```
rhcert-cli run --test ominpathconnectiontest --server <LTS IP addr>
```

Run time: This test takes less than 10 minutes to run.

Reference

See [Understanding InfiniBand and RDMA technologies](#) for more information.

A.1.1.29. pccard (Red Hat Enterprise Linux 6 only)

What the test covers: The **pccard** test covers PC Cards (also known as PCMCIA cards).

What the test does: The test uses the `/sbin/pccardctl` command to control the system's pccard sockets individually. It loops through all the sockets and performs three actions: a power off, power on and a card query to get the identity of the inserted card(s).

Preparing for the test: Each card slot must be populated before running the test. The `/sbin/pccardctl` utility has the ability to turn the slots off and on, simulating an eject and an insert, so the tester is not prompted to insert cards at test time.

Executing the test: The pccard test is non-interactive. Check the checkbox next to the test and click the **Run Selected** button to perform the test.

A.1.1.30. profiler

What the test does: The **profiler** test will attempt to shut down **oprofile** to get a clean slate, then bring it up correctly (start the daemon). It will load all the oprofile modules and a handful of additional support items (for example, some directories under `/dev` are mounted), then it will start the oprofile application. The application will acquire some sample data, called a report, then quit. If all those steps are completed successfully, the test passes. There is another loop in the test that is executed if one of those actions fails. The oprofile application requires specific hardware registers in the CPU to record its data. If for some reason this dedicated support is not working (or the hardware counters are not present), the other loop enables timer mode, allowing the data to be recorded in software instead of in the CPU registers. If you encounter failures in the profiler test, try forcing timer mode by adding this line to `/etc/modprobe.conf` and then rebooting before attempting to run the test again:

```
options oprofile timer=1
```

Preparing for the test: To prepare for the test, ensure that a CPU is installed that meets the requirements explained in the Policy Guide.

Executing the test: The profiler test is non-interactive. Check the checkbox next to the test and click the **Run Selected** button to perform the test.

Run time: The profiler test takes approximately 30 seconds on a 2013-era workstation. The required info test will add about a minute to the overall run time.

A.1.1.31. realtime

**NOTE**

This test only runs when certifying hardware on the Red Hat Enterprise Linux for Real Time product on Red Hat Enterprise Linux 7.

What the test covers: The **realtime** test covers the testing of systems running Red Hat Enterprise Linux for Real Time with two sets of tests: one to find system management mode-based execution delays, and one to determine the latency of servicing timer events.

What the test does: The first portion of the test loads a special kernel module named **hwlat_detector.ko**. This module creates a kernel thread which polls the Timestamp Counter Register (TSC), looking for intervals between consecutive reads which exceed a specified threshold. Gaps in consecutive TSC reads mean that the system was interrupted between the reads and executed other code, usually System Management Mode (SMM) code defined by the system BIOS.

The second part of the test starts a program named **cyclictest**, which starts a measurement thread per cpu, running at a high realtime priority. These threads have a period (100 microseconds) where they perform the following calculation:

1. get a timestamp (t1)
2. sleep for period
3. get a second timestamp (t2)
4. $\text{latency} = t2 - (t1 + \text{period})$
5. goto 1

The latency is the time difference between the theoretical wakeup time (t1+period) and the actual wakeup time (t2). Each measurement thread tracks minimum, maximum and average latency as well as reporting each datapoint.

Once **cyclictest** is running, **rteval** starts a pair of system loads, one being a parallel linux kernel compile and the other being a scheduler benchmark called **hackbench**.

When the run is complete, **rteval** performs a statistical analysis of the data points, calculating mean, mode, median, variance and standard deviation.

Preparing for the test: Install and boot the realtime kernel-rt kernel before adding the system to the certification. The command will detect that the running kernel is a realtime kernel and will schedule the realtime test to be run.

Running the test: The realtime test is non-interactive. Check the checkbox next to the test and click the **Run Selected** button to perform the test. The test will only appear when the system is running the rt-kernel.

Run time: The system management mode portion of the test runs for two hours. The timer event analysis portion of the test runs for twelve hours on all machines. The required info test will add about a minute to the overall run time.

A.1.1.32. reboot (Optional)

What the test covers: The **reboot** test confirms the ability of a system to reboot when prompted. It is not required for certification at this time.

What the test does: The test issues a **shutdown -r 0** command to immediately reboot the system with no delay.

Preparing for the test: Ensure that the system can be rebooted before running this test by closing any running applications.

Executing the test: The reboot test is interactive. Check the box next to the test name to indicate it is among the tests to run. Click the button **Run Selected** to continue. You will be asked **Ready to restart?** when you reach the reboot portion of the test program. Answer y if you are ready to perform the test. The system will reboot and after coming back up, the test server will verify that the reboot completed successfully.

A.1.1.33. RoCE connection

What the test does: The Omnipath Connection test runs the following subtests to ensure a baseline functionality using, when appropriate, the ip address selected from the dropdown at the onset of the test:

1. Ping test - Runs ping from the starting IP address of the device being tested on the SUT to the selected IP address of the LTS.
2. Rping test - Runs rping on LTS and SUT using the selected LTS IP address, then compares results to verify it ran to completion.
3. Rcopy test - Runs rcopy on LTS and SUT, sending a randomly generated file and comparing md5sums on LTS and SUT to verify successful transfer.
4. Ethtool test - Runs the ethtool command passing in the detected net device of the roce device.

Preparing for the test: Ensure that the LTS and SUT are separate machines, on the same fabric(s).

Executing the test: This is an interactive test. Check the box next to the test name to indicate it is among the tests to run. Click the button **Run Selected** to continue. You will be prompted with a dropdown to select an ip address (an ip address of LTS) in which to perform the tests using. Select an ip address corresponding to a device on the same fabric of the SUT device you are running the test for.

Manually adding and running the test: Use the following command to manually add the RoCEConnectionTest:

```
rhcert-cli plan --add --test RoCEConnectionTest --device <device
name>_devicePort_<port number>_netDevice_<net device here>
```

Use the following command to manually run the RoCEConnectionTest:

```
rhcert-cli run --test RoCEConnectionTest --server <LTS IP addr>
```

Run time: This test takes less than 10 minutes to run.

Reference

See [Understanding InfiniBand and RDMA technologies](#) for more information.

A.1.1.34. SATA

What the SATA test covers:

There are many different kinds of persistent on-line storage devices available in systems today.

What the test does:

The **SATA** test is designed to test anything that reports an **ID_TYPE** of "disk" in the udev database. This test is for **SATA drives**. The hwcert/storage/SATA test gets planned if:

- the controller name of any disk mentions **SATA**, or
- the lsscsi transport for the host that disks are connected to mentions **SATA**

If the above two criteria do not meet, then the storage test would get planned for the detected device.

For more information on **what the test does** and **preparing for the test** see [Section A.1.1.44, "STORAGE"](#)

A.1.1.35. SATA_SSD**What the SATA_SSD test covers:**

This test will run if it determines the storage unit of interest is SSD and its interface is SATA.

What the SATA_SSD test does:

The test finds the SCSI storage type and identifies connected storage interface on the location more `/sys/block/sdap/queue/rotational`. The test is planned if the rotational bit is set to zero for SSD.

Following are the device parameter values that would be printed as part of the test:

- `logical_block_size` - Used to address a location on the device
- `physical_block_size` - Smallest unit on which the device can operate
- `minimum_io_size` - Minimum unit preferred for random input/output of device's
- `optimal_io_size` - It is the preferred unit of device's for streaming input/output
- `alignment_offset` - It is offset value from the underlying physical alignment

For more information on **what the test does** and **preparing for the test** see [Section A.1.1.44, "STORAGE"](#)

A.1.1.36. M2_SATA**What the M2_SATA test covers:**

This test will run if it determines the interface is SATA and attached through an M2 connection.

Manually adding and running the test:

Use the following command to manually add the M2_SATA test:

```
rhcert-cli plan --add --test M2_SATA --device host0
```

Following are the device parameter values that would be printed as part of the test:

- `logical_block_size` - Used to address a location on the device

- `physical_block_size` - Smallest unit on which the device can operate
- `minimum_io_size` - Minimum unit preferred for random input/output of device's
- `optimal_io_size` - It is the preferred unit of device's for streaming input/output
- `alignment_offset` - It is offset value from the underlying physical alignment

For more information on **what the test does** and **preparing for the test** see [Section A.1.1.44](#), "STORAGE"

A.1.1.37. U2_SATA

What the U2_SATA test covers:

This test will run if it determines the interface is SATA and attached through a U2 connection.

Manually adding and running the test:

Use the following command to manually add the U2_SATA test:

```
rhcert-cli plan --add --test U2_SATA --device host0
```

Following are the device parameter values that would be printed as part of the test:

- `logical_block_size` - Used to address a location on the device
- `physical_block_size` - Smallest unit on which the device can operate
- `minimum_io_size` - Minimum unit preferred for random input/output of device's
- `optimal_io_size` - It is the preferred unit of device's for streaming input/output
- `alignment_offset` - It is offset value from the underlying physical alignment

For more information on **what the test does** and **preparing for the test** see [Section A.1.1.44](#), "STORAGE"

A.1.1.38. SAS

What the SAS test covers:

There are many different kinds of persistent on-line storage devices available in systems today.

What the test does:

The **SAS** test is designed to test anything that reports an **ID_TYPE** of "disk" in the udev database. This test is for **SAS drives**. The `hwcert/storage/SAS` test gets planned if:

- the controller name of any disk should mention **SAS**, or
- the `lsscsi` transport for the host that disks are connected to should mention **SAS**

If the above two criteria do not meet, then the storage test would get planned for the detected device.

For more information on **what the test does** and **preparing for the test** see [Section A.1.1.44](#), "STORAGE"

A.1.1.39. SAS_SSD

What the SAS_SSD test covers:

This test will run if it determines the storage unit of interest is SSD and its interface is SAS.

What the SAS_SSD test does:

The test finds the SCSI storage type and identifies connected storage interface on the location more /sys/block/sdap/queue/rotational. The test is planned if the rotational bit is set to zero for SSD.

Following are the device parameter values that are printed as part of the test:

- logical_block_size - Used to address a location on the device
- physical_block_size - Smallest unit on which the device can operate
- minimum_io_size - Minimum unit preferred for random input/output of device's
- optimal_io_size - It is the preferred unit of device's for streaming input/output
- alignment_offset - It is offset value from the underlying physical alignment

For more information on **what the test does** and **preparing for the test** see [Section A.1.1.44, "STORAGE"](#)

A.1.1.40. PCIE_NVMe

What the PCIE_NVMe test covers:

This test will run if it determines the interface is NVMe and attached through a PCIE connection.

What the PCIE_NVMe test does:

This test gets planned if logical device host name string contains " nvme[0-9] "

Following are the device parameter values that would be printed as part of the test:

- logical_block_size - Used to address a location on the device
- physical_block_size - Smallest unit on which the device can operate
- minimum_io_size - Minimum unit preferred for random input/output of device's
- optimal_io_size - It is the preferred unit of device's for streaming input/output
- alignment_offset - It is offset value from the underlying physical alignment

For more information on **what the test does** and **preparing for the test** see [Section A.1.1.44, "STORAGE"](#)

A.1.1.41. M2_NVMe

What the M2_NVMe test covers:

This test will run if it determines the interface is NVMe and attached through an M2 connection.

Manually adding and running the test:

Use the following command to manually add the M2_NVMe test:

```
rhcert-cli plan --add --test M2_NVMe --device nvme0
```

Following are the device parameter values that would be printed as part of the test:

- `logical_block_size` - Used to address a location on the device
- `physical_block_size` - Smallest unit on which the device can operate
- `minimum_io_size` - Minimum unit preferred for random input/output of device's
- `optimal_io_size` - It is the preferred unit of device's for streaming input/output
- `alignment_offset` - It is offset value from the underlying physical alignment

For more information on **what the test does** and **preparing for the test** see [Section A.1.1.44, "STORAGE"](#)

A.1.1.42. U2_NVMe**What the U2_NVMe test covers:**

This test will run if it determines the interface is NVMe and attached through a U2 connection.

Manually adding and running the test:

Use the following command to manually add the U2_NVMe test:

```
rhcert-cli plan --add --test U2_NVMe --device nvme0
```

Following are the device parameter values that would be printed as part of the test:

- `logical_block_size` - Used to address a location on the device
- `physical_block_size` - Smallest unit on which the device can operate
- `minimum_io_size` - Minimum unit preferred for random input/output of device's
- `optimal_io_size` - It is the preferred unit of device's for streaming input/output
- `alignment_offset` - It is offset value from the underlying physical alignment

For more information on **what the test does** and **preparing for the test** see [Section A.1.1.44, "STORAGE"](#)

[[sect- NVDIMM]]

A.1.1.43. NVDIMM**What the NVDIMM test covers:**

This test operates like any other SSD non-rotational storage test and identifies the NVDIMM storage devices

What the test does:

The test gets planned for storage device if:

- There exist namespaces (non-volatile memory devices) for that disk device reported by "ndctl list"
- It reports the "DEVTYPE" of the sda is equal to 'disk'

Following are the device parameter values that would be printed as part of the test:

- logical_block_size - Used to address a location on the device
- physical_block_size - Smallest unit on which the device can operate
- minimum_io_size - Minimum unit preferred for random input/output of device's
- optimal_io_size - It is the preferred unit of device's for streaming input/output
- alignment_offset - It is offset value from the underlying physical alignment

For more information on **what the test does** and **preparing for the test** see [Section A.1.1.44, "STORAGE"](#)

A.1.1.44. STORAGE

What the storage test covers: There are many different kinds of persistent on-line storage devices available in systems today. The **STORAGE** test is designed to test anything that reports an **ID_TYPE** of "disk" in the udev database. This includes IDE, SCSI, SATA, SAS, and SSD drives, PCIe SSD block storage devices, as well as SD media, xD media, MemoryStick and MMC cards. The test plan script reads through the udev database and looks for storage devices that meet the above criteria. When it finds one, it records the device and its parent and compares it to the parents of any other recorded devices. It does this to ensure that only devices with unique parents are tested. If the parent has not been seen before, the device is added to the test plan. This speeds up testing as only one device per controller will be tested, as per the Policy Guide.

What the test does: The STORAGE test performs the following actions on all storage devices with a unique parent:

1. The script looks through the partition table to locate a swap partition that is not on an LVM or software RAID device. If found, it will deactivate it with **swaponoff** and use that space for the test. If no swap is present, the system can still test the drive if it is completely blank (no partitions). Note that the swap device must be active in order for this to work (the test reads **/proc/swaps** to find the swap partitions) and that the swap partition must not be inside any kind of software-based container (no LVM or software RAID, but hardware RAID would work as it would be invisible to the system).
2. The tool creates a filesystem on the device, either in a swap partition on the blank drive.
3. The filesystem is mounted and **dt** is used to test the device. The dt command is the "data test" program and is a generic test tool capable of testing reads and writes to devices (among other things).
4. After the mounted filesystem test, the filesystem is unmounted and a dt test is performed against the block device, ignoring the file system. The dt test uses the "direct" parameter to handle this.

Preparing for the test: You should install all the drives and storage controllers that are listed on the

official test plan. In the case of multiple storage options, as many as can fit into the system at one time can be tested in a single run, or each storage device can be installed individually and have its own run of the storage test. You can decide on the order of testing and number of controllers present for each test. Each logical drive attached to the system must contain a swap partition in addition to any other partitions, or be totally blank. This is to provide the test with a location to create a filesystem and run the tests. The use of swap partitions will lead to a much quicker test, as devices left blank are tested in their entirety. They will almost always be significantly larger than a swap partition placed on the drive. Please see the Red Hat Knowledgebase article at <https://access.redhat.com/site/solutions/15244> for more information on appropriate swap file sizing.



NOTE

If testing an SD media card, use the fastest card you can obtain. While a Class 4 SD card may take 8 hours or more to run the test, a Class 10 or UHS 1/2 card can complete the test run in 30 minutes or less.

When it comes to choosing storage devices for the official test plan, the rule that the review team operates by is "one test per code path". What we mean by that is that we want to see a storage test run using every driver that a controller can use. The scenario of multiple drivers for the same controller usually involves RAID storage of some type. It's common for storage controllers to use one driver when in regular disk mode and another when in RAID mode. Some even use multiple drivers depending on the RAID mode that they are in. The review team will analyze all storage hardware to determine the drivers that need to be used in order to fulfill all the testing requirements. That's why you may see the same storage device listed more than once in the official test plan. Complete information on storage device testing is available in the Policy Guide.

Executing the test: The storage test is non-interactive. Check the checkbox next to the test and click the **Run Selected** button to perform the test.

```
Host bus adapter host0 has storage devices sda, sda1, sda2, sda3
Which disk would you like to test: (sda|sda1|sda2|sda3|all)
```

Run time, bare-metal: The storage test takes approximately 22 minutes on a 6Gb/s SATA hard drive installed in a 2013-era workstation system. The same test takes approximately 3 minutes on a 6Gb/s SATA solid-state drive installed in a 2013-era workstation system. The required info test will add about a minute to the overall run time.

A.1.1.45. suspend (Laptops only)

What the test covers: The **suspend** test covers suspend/resume from S3 sleep state (suspend to RAM) and suspend/resume from S4 hibernation (suspend to disk). This test is only scheduled on systems that have built-in batteries, like laptops, so it won't be present on any other type of system.



IMPORTANT

The suspend to RAM and suspend to disk abilities are essential characteristics of laptops. We therefore schedule an automated suspend test at the beginning of all certification test runs on a laptop. This ensures that all hardware functions normally post-resume. The test will always run on a laptop, much like the info test, regardless of what tests are scheduled.

What the test does: The test queries the `/sys/power/state` file and determines which states are supported by the hardware. If it sees "mem" in the file, it schedules the S3 sleep test. If it sees "disk" in the file, it schedules the S4 hibernation test. If it sees both, it schedules both. What follows is the

procedure for a system that supports both S3 and S4 states. If your system does not support both types it will only run the tests related to the supported type.

- If S3 sleep is supported, the script uses the **pm-suspend** command to suspend to RAM. The tester wakes the system up after it sleeps and the scripts check the exit code of **pm-suspend** to verify that the system woke up correctly. Testing then continues on the test server interface.
- If S4 hibernation is supported, the script uses the use the **pm-suspend** command to suspend to disk. The tester wakes the system up after it hibernates and the scripts check the exit code of **pm-suspend** to verify that the system woke up correctly. Testing then continues on the test server interface.
- If S3 sleep is supported, the tester is prompted to press the key that manually invokes it (a **Fn+F-key** combination or dedicated **Sleep** key) if such a key is present. The tester wakes the system up after it sleeps and the scripts check the exit code of **pm-suspend** to verify that the system woke up correctly. Testing then continues on the test server interface. If the system has no suspend key, this section can be skipped.
- If S4 hibernation is supported, the tester is prompted to press the key that manually invokes it (a **Fn+F-key** combination or dedicated **Hibernate** key) if such a key is present. The tester wakes the system up after it hibernates and the scripts check the exit code of **pm-suspend** to verify that the system woke up correctly. Testing then continues on the test server interface. If the system has no suspend key, this section can be skipped.

Preparing for the test: Ensure that a swap file large enough to hold the contents of RAM was created when the system was installed. Guidelines for swap file size can be found at this Red Hat Knowledgebase article: <https://access.redhat.com/site/solutions/15244>. Also, someone must be present at the system under test in order to wake it up from suspend and hibernate.

Executing the test: The suspend test is interactive. Check the box next to the test name to indicate it is among the tests to run. Click the button Run Selected to continue. The test server GUI will display a status of *suspend?* when the test runs. Click on the *suspend?* status link or the **Continue Testing** button and then click the **Yes** button to suspend the laptop.

The test server will display *waiting for response* after it sends the suspend command. Check the laptop and confirm that it has completed suspending, then press the power button or any other key that will wake it from suspend. The test server will continuously monitor the system under test to see if it has awakened. Once it has woken up, the test server GUI will display the question *Has resume completed?*. Press the **Yes** or **No** button to tell the test server what happened.

The server will then continue to the hibernate test. Again, click the **Yes** button under the *suspend?* question to put the laptop into hibernate mode.

The test server will display *waiting for response* after it sends the hibernate command. Check the laptop and confirm that it has completed hibernating, then press the power button or any other key that will wake it from hibernation. The test server will continuously monitor the system under test to see if it has awakened. Once it has woken up, the test server GUI will display the question *has resume completed?*. Press the **Yes** or **No** button to tell the test server what happened.

Next the test server will ask you if the system has a keyboard key that will cause the system under test to suspend. If it does, click the **Yes** button under the question *Does this system have a function key (Fn) to suspend the system to mem?*. Follow the procedure described above to verify suspend and wake the system up to continue with testing.

Finally the test server will ask you if the system has a keyboard key that will cause the system under test to hibernate. If it does, click the **Yes** button under the question *Does this system have a function key*

(Fn) to suspend the system to disk? Follow the procedure described above to verify hibernation and wake the system up to continue with any additional tests you have scheduled.

Run time: The suspend test takes about 6 minutes on a 2012-era laptop with 4GB of RAM and a non-SSD hard drive. This is the time for a full series of tests, including both pm-suspend-based and function-key-based suspend and hibernate runs. The time will vary depending on the speed at which the laptop can write to disk, the amount and speed of the RAM installed, and the capability of the laptop to enter suspend and hibernate states through function keys. The required info test will add about a minute to the overall run time.

A.1.1.46. tape

What the test covers: The **tape** test covers all types of tape drives. Any robots associated with the drives are not tested by this test.

What the test does: The test uses the **mt** command to rewind the tape, then it does a **tar** of the **/usr** directory and stores it on the tape. A **tar** compare is used to determine if the data on the tape matches the data on the disk. If the data matches, the test passes.

Preparing for the test: Insert a tape of the appropriate size into the drive.

Executing the test: The tape test is non-interactive. Check the checkbox next to the test and click the **Run Selected** button to perform the test.

A.1.1.47. USB2

What the test covers: The **USB2** test covers USB2 ports from a basic functionality standpoint, ensuring that all ports can be accessed by the OS.

What the test does: The purpose of the test is to ensure that all USB2 ports present in a system function as expected. It asks for the number of available USB2 ports (minus any that are in use for keyboard/mouse, etc.) and then asks the tester to plug and unplug a USB2 device into each port. The test watches for attach and detach events and records them. If it detects both plug and unplug events for the number of unique ports the tester entered, the test will pass.

Preparing for the test: Count the available USB2 ports and have a spare USB2 device available to use during the test. You may need to trace the USB ports from the motherboard header(s) to distinguish between USB2 and USB3 ports.

Executing the test: The USB2 test is interactive. Check the box next to the test name to indicate it is among the tests to run. Click the button Run Selected to continue. When prompted by the system, enter the number of available USB2 ports present on the system. Don't count any that are currently in use by keyboards or mice. The system will ask for the test USB2 device to be plugged into a port and will then pause until the tester presses *y* to continue. The system will then ask for the device to be unplugged and again will pause until the tester presses *y* to continue. These steps repeat for the number of ports that were entered. Note that there is no right or wrong order for testing the ports, but each port must be tested only once.

Run time: The USB2 test takes about 15 seconds per USB2 port. This includes the time to manually plug in the device, scan the port, unplug the device, and scan the port again. The required info test will add about a minute to the overall run time.

A.1.1.48. USB3

What the test covers: The **USB3** test covers USB3 ports from a basic functionality standpoint, ensuring that all ports can be accessed by the OS.

What the test does: The purpose of the test is to ensure that all USB3 ports present in a system function as expected. It asks for the number of available USB3 ports (minus any that are in use for keyboard/mouse, etc.) and then asks the tester to plug and unplug a USB3 device into each port. The test watches for attach and detach events and records them. If it detects both plug and unplug events for the number of unique ports the tester entered, the test will pass.

Preparing for the test: Count the available USB3 ports and have a spare USB3 device available to use during the test. You may need to trace the USB ports from the motherboard header(s) to distinguish between USB2 and USB3 ports.

Executing the test: The USB3 test is interactive. Check the box next to the test name to indicate it is among the tests to run. Click the button **Run Selected** to continue. When prompted by the system, enter the number of available USB3 ports present on the system. Don't count any that are currently in use by keyboards or mice. The system will ask for the test USB3 device to be plugged into a port and will then pause until the tester presses *y* to continue. The system will then ask for the device to be unplugged and again will pause until the tester presses *y* to continue. These steps repeat for the number of ports that were entered. Note that there is no right or wrong order for testing the ports, but each port must be tested only once.

Run time: The USB3 test takes about 15 seconds per USB3 port. This includes the time to manually plug in the device, scan the port, unplug the device, and scan the port again. The required info test will add about a minute to the overall run time.

A.1.1.49. video

What the test covers: All video hardware, whether removeable or integrated on the motherboard, is tested using the **video** test. Devices are selected for testing by their PCI class ID. Specifically, the test is looking for a device class of "30000" in the output of `udev`.

What the test does: The video test first determines which command is used to control the X configuration on the machine where it is running (either **redhat-config-xfree86** or **system-config-display**). It then runs it with the **--noui** flag and generates a clean X configuration file. It runs **startx** using the new configuration file and runs **x11perf**, which is a X11 server performance test program. After the performance test completes it also runs **xdpyinfo** to determine the screen resolution and color depth. The configuration file created at the start of the test should allow the system to run at the maximum resolution that the monitor and video card are capable of achieving. The final portion of the test uses **grep** to search through the `/var/log/Xorg.0.log` logfile to determine which driver is being used.

Preparing for the test: Ensure that the monitor and video card in the system are capable of running at a resolution of 1024x768 with a color depth of 24 bits per pixel (bpp). This is the minimum resolution and color depth required to achieve a passing video test. Higher resolutions or color depths are also acceptable, but nothing lower than 1024x768 at 24bpp will pass. You can confirm this ability by looking at the output of **xrandr**. All the resolutions that can be achieved by the monitor and video card should be displayed in the output of **xrandr**. Check the output for 1024x768 at 24 bits per pixel (or higher). You may need to remove any KVM switches that are between the monitor and video card if you are not seeing all the resolutions that the card/monitor combination are capable of generating.

Executing the test: The video test is non-interactive. Check the checkbox next to the test and click the **Run Selected** button to perform the test. The screen on the test system will go blank, followed by a series of test patterns from the **x11perf** test program. It will return to the desktop or to the virtual terminal screen that the system was on at execution time when the test finishes.

Run time: The video test takes about 1 minute to perform on a 2013-era workstation. The required info test will add about a minute to the overall run time.

A.1.1.50. WirelessG

What the test covers: The **WirelessG** test is run on all wireless Ethernet connections with a maximum connection speed of 802.11g.

What the test does: This is a new test that combines the existing **wlan** and **network** tests. In addition to passing all the existing network test items, this test must detect a "g" link type as reported by **iw** and demonstrate a throughput of 22Mb/s (with a margin for overhead) in order to pass. Please see [Section A.1.1.27, "network"](#) for information on the rest of the test functionality.

A.1.1.51. WirelessN

What the test covers: The **WirelessN** test is run on all wireless Ethernet connections with a maximum connection speed of 802.11n.

What the test does: This is a new test that combines the existing **wlan** and **network** tests. In addition to passing all the existing network test items, this test must detect an "n" link type as reported by **iw** and demonstrate a throughput of 100Mb/s (with a margin for overhead) in order to pass. Please see [Section A.1.1.27, "network"](#) for information on the rest of the test functionality.

A.1.1.52. WirelessAC (Red Hat Enterprise Linux 7 only)



NOTE

The WirelessAC test will not plan automatically at this time, as we are waiting for full 802.11ac support to be incorporated into Red Hat Enterprise Linux. All 802.11ac-capable systems will have the WirelessN test planned instead, and only "N" speeds are required to pass the test.

What the test covers: The **WirelessAC** test is run on all wireless Ethernet connections with a maximum connection speed of 802.11ac.

What the test does: This is a new test that combines the existing **wlan** and **network** tests. In addition to passing all the existing network test items, this test must detect an "ac" link type as reported by **iw** and demonstrate a throughput of 300Mb/s (with a margin for overhead) in order to pass. Please see [Section A.1.1.27, "network"](#) for information on the rest of the test functionality.

A.1.2. Manually Adding Tests

On rare occasions, tests may fail to plan due to problems with hardware detection or other issues with the hardware, OS, or test scripts. If this happens you should get in touch with your Red Hat support contact for further assistance. They will likely ask you to open a support ticket for the issue, and then explain how to manually add a test to your local test plan using the **rhcert-cli** command on the SUT. Any modifications you make to the local test plan will be sent to the LTS, so you can continue to use the web interface on the LTS to run your tests. The command is run as follows:

```
# rhcert-cli plan --add --test=<testname> --device=<devicename> --udi-
<udi>
```

The options for the **rhcert-cli** command used here are:

- **plan** - Modify the test plan
- **--add** - Add an item to the test plan

- **--test=<testname>** - The test to be added. The test names are as follows:
 - hwcert/suspend
 - hwcert/audio
 - hwcert/battery
 - hwcert/lid
 - hwcert/usbbase/expresscard
 - hwcert/usbbase/usbbase/usb2
 - hwcert/usbbase/usbbase/usb3
 - hwcert/kdump
 - hwcert/network/Ethernet/100MegEthernet
 - hwcert/network/Ethernet/1GigEthernet
 - hwcert/network/Ethernet/10GigEthernet
 - hwcert/network/Ethernet/40GigEthernet
 - hwcert/network/wlan/WirelessG
 - hwcert/network/wlan/WirelessN
 - hwcert/network/wlan/WirelessAC (available in Red Hat Enterprise Linux 7 only)
 - hwcert/memory
 - hwcert/core
 - hwcert/cpuscaling
 - hwcert/fvtest/fv_core
 - hwcert/fvtest/fv_memory
 - hwcert/fvtest/fv_network
 - hwcert/fvtest/fv_storage
 - hwcert/profiler
 - hwcert/storage
 - hwcert/video
 - hwcert/info
 - hwcert/optical/bluray
 - hwcert/optical/dvd

- hwcert/optical/cdrom
- hwcert/fencing
- hwcert/realtime
- hwcert/reboot
- hwcert/tape
- The other options are only needed if a device must be specified, like in the network and storage tests that need to be told which device to run on. There are various places you would need to look to determine the device name or UDI that would be used here. Support can help determine the proper name or UDI. Once found, you would use one of the following two options to specify the device:
 - **--device=<devicename>** - The device that should be tested, identified by a device name such as "enp0s25" or "host0".
 - **--udi=<UDI>** - The unique device ID of the device to be tested, identified by a UDI string.

Revised on 2018-11-01 11:42:14 UTC