Red Hat Hardware Certification 2024

Red Hat Hardware Certification Test Suite User Guide

For Use with Red Hat Hardware Certification

Last Updated: 2024-06-24

Red Hat Hardware Certification 2024 Red Hat Hardware Certification Test Suite User Guide

For Use with Red Hat Hardware Certification

Legal Notice

Abstract

The Red Hat Hardware 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, explains how to set up the certification environment, test the systems or components being certified, and submit the results to Red Hat for verification. The guide also provides the background information including the test methodology and results evaluation. Version 9.0 and 8.80 updated May 28, 2024.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

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

CHAPTER 1. INTRODUCTION TO RED HAT HARDWARE CERTIFICATION PROGRAM

Use this guide to certify your company's hardware products to run one or more of Red Hat's products.

1.1. THE RED HAT CERTIFICATION PROGRAM OVERVIEW

The Red Hat Certification Program ensures compatibility of Red Hat's partner's hardware and software products with Red Hat Enterprise Linux, Red Hat OpenStack Platform, Red Hat Enterprise Linux for Real Time, and other Red Hat software products on the Hardware Platform. The program has three main elements:

- **Test suite**: Comprises tests for hardware or software applications undergoing certification.
- **Red Hat Certification Ecosystem** Helps to explore and find certified products including hardware, software, cloud, and service providers.
- **Support**: A joint support relationship between you and Red Hat.

1.2. CERTIFICATION WORKFLOW

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

Prerequisites

1. Establish a certification relationship with Red Hat.
2. Set up a test environment consisting of the partner's product and the Red Hat product combination to be certified.
3. Do preliminary testing to ensure this combination works well.
4. Install the redhat-certification tool.

Procedure

1. Create a certification request for a specific software or hardware component using the Red Hat Certification tool.
2. The Red Hat certification team applies certification policies to the hardware specifications to create the official test plan. Test plans to certify systems or components for RHEL 8 and RHEL 9 consist of tests and features that will be published based on the identified components and their specifications submitted to Red Hat.
3. Run the tests specified in the official test plan and submit the results by using the Red Hat Certification tool to the Red Hat certification team for analysis.
4. The certification team analyzes the test results and communicates any required retesting.
5. Provide Red Hat with a representative hardware sample that covers the items that are being certified.

6. When all tests have favorable results, the certification is complete and the certified product is made available on the Red Hat Ecosystem Catalog .

Additional resources

- For more information about requirements and policies for Red Hat hardware certification, see Red Hat Hardware Certification Policy Guide

1.3. GETTING SUPPORT AND GIVING FEEDBACK

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 or problem faced in the certification process or Red Hat product or feedback on the certification toolset or documentation.

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

Complete the remaining form using the details How do I open and manage a support case on the Customer Portal?

NOTE

Red Hat recommends that you are a Red Hat Certified Engineer or hold equivalent experience before starting the certification process.

CHAPTER 2. PARTNER ONBOARDING

Use the Red Hat Customer Portal to create a new account and to join the hardware certification program.

If you face any issues during the certification process, you can contact us for support in any of the following ways.

- Create a Partner Acceleration Desk (PAD) ticket under the **Product Certification** category.
- Email the Red Hat Certification Operation (cert-ops) team at cert-ops@redhat.com.
- Contact your dedicated Ecosystem Partner Management (EPM) if you have been assigned one.

2.1. CREATING A RED HAT ACCOUNT

Procedure

1. Open the Red Hat Customer Portal and click **Register** on the top-right corner of the page. The **Register for a Red Hat account**page displays.

NOTE

Make sure you use your company email ID and not your personal email. The email ID you enter will be used for all your communication with Red Hat going forward.

NOTE

Red Hat recommends using a unique login ID that is separate from the email ID to avoid account-related issues in the future. The login ID, once created, cannot be changed.

2. Enter your **Login information** and **Personal information**.
3. Select **Corporate** as the Account type.
4. Enter your company's **Contact information**.
5. Click **Create My Account** A new Red Hat account is created.
6. Request to enable Red Hat Partner Subscription (RHPS) on the same account. You must have an administrator organization (Org. Admins) rights to make a request.
   See Red Hat Partner Subscription (RHPS) for instructions.

Verification

- Verify that an account number is assigned to your account.
  To do this, log in to your account at the Red Hat Customer Portal , and then click your avatar at the top-right to confirm the account details.

2.2. JOIN THE HARDWARE CERTIFICATION PROGRAM

Procedure

1. Log in to the Red Hat Partner Connect portal to join the hardware certification program.
2. Click **Accept Terms and Conditions**.
3. Provide some more information about your company and product, and click **Submit**.
4. Verify your email address by clicking on the link you received in your mailbox.
5. On the **Red Hat Terms and Conditions** page, select all the **I have read and agree to the terms** check boxes, and click **Submit**.
   A message is displayed on successfully becoming a hardware partner.
6. Select an option from the **How will you use this subscription?** drop-down list, and click **Request partner subscription**.
7. On the **Red Hat Terms and Conditions** page, select all the **I have read and agree to the terms** check boxes, and click **Submit**.
   A message is displayed after successfully receiving a free partner subscription.

   A vendor profile is created and Single Sign-on (SSO) is automatically added to the vendor's users.

NOTE

Your vendor profile, once created, remains in the Red Hat database. However, your Red Hat Partner Subscription (RHPS) account becomes inactive after one year of inactivity. Therefore, if you are a returning partner, raise a request to reactivate your RHPS account before beginning hardware certification.

Upon activating your RHPS account, the cert-ops team will be notified of your details, including Single Sign-on (SSO) information assigned to your account.

Next Steps

Opening a new certification case using the Red Hat Certification Tool

CHAPTER 3. OPENING A NEW CERTIFICATION CASE USING THE RED HAT CERTIFICATION TOOL

As a partner, you can start the certification process by opening a new certification case in the Red Hat Certification Tool. When you complete this task, Red Hat prepares a test plan based on the product specification you provide.

Prerequisites

- You have established a certification relationship with Red Hat.
- You have the user login credentials.
- You have the vendor and products linked to your user login.

Procedure

1. Log in to Red Hat Certification Portal.
2. On the home page, click **Open Certification**.
   The **Open a New Certification Case** dialog opens.
3. Click **Next**.
4. Select an option from the **Partner** and **Product** list.
   If your product does not appear, create it by entering its name in the **Product** field. Then, select it.
5. In the **What kind of product is this?** section, select the checkbox applicable to your product.
   Your product might qualify for more than one ecosystem.
6. Click **Next**.
7. Enter the **Make**.
   The **Model** is displayed based on the previously entered information.
8. Select the applicable checkbox under **Which category best describes your product?**
9. Optional: Enter the **Product URL**, **Support URL**, and **Specification URL**.
10. Click **Next**.
    Based on your inputs, a new product is created in the **Partner Product** list.
11. Select an option from the **Red Hat Certification** list and click **Next**.
12. Review the information you provided and click **Next**.

Verification

If you have successfully created a new certification case for your product, a message is displayed.

Next steps

While Red Hat prepares a test plan for the product, you can Set up the test environment to prepare the systems for running tests.

CHAPTER 4. SETTING UP THE TEST ENVIRONMENT

Set up the test environment where you can run the tests.

The test environment consists of at least two systems:

- System 1: Acts as the host under test (HUT)
- System 2: Acts as the test server

4.1. SETTING UP THE HOST UNDER TEST

A system on which the product that needs certification is installed or configured is referred to as the host under test (HUT).

Prerequisites

- The HUT has RHEL version 8 or 9 installed. For convenience, Red Hat provides kickstart files to install the HUT's operating system. Follow the instructions in the file that is appropriate for your system before launching the installation process.

NOTE

Red Hat Hardware Certification requires using the General Availability (GA) kernel for the Red Hat Enterprise Linux version being certified for hardware certification.

When installing RHEL, please download and use the Binary DVD Offline Install Image, and not the Boot ISO image. The Boot ISO image requires a network connection, and will automatically install the current kernel instead of the required GA kernel.

Do not register the system with Red Hat Subscription Management (RHSM) during the installation process. Only register the system with RHSM after the installation process is completed.

Procedure

1. Configure the *Red Hat Certification* repository.
   Use your RHN credentials to register your system using Red Hat Subscription Management:

   ```
   # subscription-manager register
   ```

2. Display the list of available subscriptions for your system:

   ```
   # subscription-manager list --available*
   ```

3. Search for the subscription which provides the Red Hat Certification (for RHEL Server) repository and make a note of the subscription and its Pool ID.

4. Attach the subscription to your system. Replace the pool_ID with the Pool ID of the subscription.

   ```
   # subscription-manager attach --pool=<pool_ID>
   ```

NOTE

You don't have to attach the subscription to your system, if you enable the option **Simple content access for Red Hat Subscription Management**. For more details, see How do I enable Simple Content Access for Red Hat Subscription Management?

5. Subscribe to the Red Hat Certification channel:

   - On RHEL 8:

     ```
     # subscription-manager repos --enable=cert-1-for-rhel-8-_<HOSTTYPE>_-rpms
     ```

     Replace HOSTTYPE with the system architecture. To find out the system architecture, run

     ```
     uname -m
     ```

     Example:

     ```
     # subscription-manager repos --enable=cert-1-for-rhel-8-x86_64-rpms
     ```

   - On RHEL 9:

     ```
     # subscription-manager repos --enable=cert-1-for-rhel-9-_<HOSTTYPE>_-rpms
     ```

     Replace HOSTTYPE with the system architecture. To find out the system architecture, run

     ```
     uname -m
     ```

     Example:

     ```
     # subscription-manager repos --enable=cert-1-for-rhel-9-x86_64-rpms
     ```

6. Install hardware test suite package:

   ```
   # yum install redhat-certification-hardware
   ```

4.2. SETTING UP THE TEST SERVER

Some of the tests running on the host under test (HUT) require a second system to pass. This second system is known as the test server.

For example, the test that checks bandwidth, transfers data from one system to another, in order to pass.

Prerequisites

- The test server has RHEL version 8 or 9 installed. Red Hat provides kickstart files to install the HUT's operating system, but you can also use them to install the test server. Follow the instructions in the file that is appropriate for your system before launching the installation process.

Procedure

1. Configure the *Red Hat Certification* repository.
   Use your RHN credentials to register your system using Red Hat Subscription Management:

   ```
   # subscription-manager register
   ```

2. Display the list of available subscriptions for your system:

   ```
   # subscription-manager list --available*
   ```

3. Search for the subscription which provides the Red Hat Certification (for RHEL Server) repository and make a note of the subscription and its Pool ID.

4. Attach the subscription to your system. Replace the pool_ID with the Pool ID of the subscription.

   ```
   # subscription-manager attach --pool=<pool_ID>
   ```

   NOTE

   You don't have to attach the subscription to your system, if you enable the option **Simple content access for Red Hat Subscription Management**. For more details, see How do I enable Simple Content Access for Red Hat Subscription Management?

5. Subscribe to the Red Hat Certification channel:

   - On RHEL 8:

     ```
     # subscription-manager repos --enable=cert-1-for-rhel-8-<HOSTTYPE>-rpms
     ```

     Replace HOSTTYPE with the system architecture. To find out the system architecture, run

     ```
     uname -m
     ```

     Example:

     ```
     # subscription-manager repos --enable=cert-1-for-rhel-8-x86_64-rpms
     ```

   - On RHEL 9:

     ```
     # subscription-manager repos --enable=cert-1-for-rhel-9-<HOSTTYPE>-rpms
     ```

     Replace HOSTTYPE with the system architecture. To find out the system architecture, run

     ```
     uname -m
     ```

     Example:

     ```
     # subscription-manager repos --enable=cert-1-for-rhel-9-x86_64-rpms
     ```

6. Install the hardware test suite package.

```
#yum install redhat-certification-hardware
```

Next steps

See Downloading the test plan from Red Hat Certification portal .

CHAPTER 5. DOWNLOADING THE TEST PLAN FROM RED HAT CERTIFICATION PORTAL

Procedure

1. Log in to Red Hat Certification portal.
2. Search for the case number related to your product certification, and copy it.
3. Click **Cases** → enter the product case number.
4. Optional: To list the components that are tested during the test run, click **Test Plans**.
5. Click **Download Test Plan**.

Next steps

If you plan to use Cockpit to run the tests, see Configuring the systems and running tests by using Cockpit.

If you plan to use CLI to run the tests, see Configuring the systems and running tests by using CLI .

CHAPTER 6. CONFIGURING THE SYSTEMS AND RUNNING TESTS USING COCKPIT

To complete the certification process, you must configure cockpit, prepare the host under test (HUT) and test server, run the tests, and retrieve the test results.

6.1. SETTING UP THE COCKPIT SERVER

Cockpit is a RHEL tool that lets you change the configuration of your systems as well as monitor their resources from a user-friendly web-based interface.

NOTE

- You must set up Cockpit on a new system, which is separate from the host under test and test server.
- Ensure that the Cockpit has access to both the host under test and the test server.

For more information on installing and configuring Cockpit, see Getting Started using the RHEL web console on RHEL 8, Getting Started using the RHEL web console on RHEL 9 and Introducing Cockpit.

Prerequisites

- The Cockpit server has RHEL version 8 or 9 installed.
- You have installed the Cockpit plugin on your system.
- You have enabled the Cockpit service.

Procedure

1. Log in to the system where you installed Cockpit.
2. Install the Cockpit RPM provided by the Red Hat Certification team.

```
# yum install redhat-certification-cockpit
```

You must run Cockpit on port 9090.

6.2. ADDING THE HOST UNDER TEST AND THE TEST SERVER TO COCKPIT

Adding the host under test (HUT) and test server to Cockpit lets the two systems communicate by using passwordless SSH.

Repeat this procedure for adding both the systems one by one.

Prerequisites

- You have the IP address or hostname of the HUT and the test server.

Procedure

1. Enter **http://<*Cockpit_system_IP*>:9090/** in your browser to launch the Cockpit web application.
2. Enter the username and password, and then click **Login**.
3. Click the down-arrow on the logged-in cockpit user name→**Add new host**.
   The dialog box displays.
4. In the **Host** field, enter the IP address or hostname of the system.
5. In the **User name** field, enter the name you want to assign to this system.
6. Optional: Select the predefined color or select a new color of your choice for the host added.
7. Click **Add**.
8. Click **Accept key and connect** to let Cockpit communicate with the system through passwordless SSH.
9. Enter the **Password**.
10. Select the **Authorize SSH Key** checkbox.
11. Click **Log in**.

Verification

On the left panel, click **Tools →Red Hat Certification** and verify that the system you just added displays under the Hosts section on the right.

6.3. GETTING AUTHORIZATION ON THE RED HAT SSO NETWORK

Procedure

1. Enter http://<Cockpit_system_IP>:9090/ in your browser's address bar to launch the Cockpit web application.
2. Enter the username and password, and then click **Login**.
3. Select **Tools → Red Hat Certification** in the left panel.
4. On the Cockpit homepage, click **Authorize**, to establish connectivity with the Red Hat system.
   The **Log in to your Red Hat account** page displays.
5. Enter your credentials and click **Next**.
   The **Grant access to rhcert-cwe** page displays.
6. Click **Grant access**. A confirmation message displays a successful device login. You are now connected to the Cockpit web application.

6.4. DOWNLOADING TEST PLANS IN COCKPIT FROM RED HAT CERTIFICATION PORTAL

For Non-authorized or limited access users:

- To download the test plan, see Downloading the test plan from Red Hat Certification portal .

For authorized users:

Procedure

1. Enter http://<Cockpit_system_IP>:9090/ in your browser's address bar to launch the Cockpit web application.
2. Enter the username and password, and then click **Login**.
3. Select **Tools → Red Hat Certification** in the left panel.
4. Click the **Test Plans** tab. A list of **Recent Certification Support Cases** will appear.
5. Click **Download Test Plan**. A message displays confirming the successful addition of the test plan.
6. The downloaded test plan will be listed under the **File Name** of the **Test Plan Files** section.

6.5. USING THE TEST PLAN TO PREPARE THE HOST UNDER TEST FOR TESTING

Provisioning the host under test performs a number of operations, such as setting up passwordless SSH communication with the cockpit, installing the required packages on your system based on the certification type, and creating a final test plan to run, which is a list of common tests taken from both the test plan provided by Red Hat and tests generated on discovering the system requirements.

For instance, required hardware packages are installed if the test plan is designed for certifying a hardware product.

Prerequisites

- You have downloaded the test plan provided by Red Hat .

Procedure

1. Enter **http://<*Cockpit_system_IP*>:9090/** in your browser address bar to launch the Cockpit web application.
2. Enter the username and password, and then click **Login**.
3. Select **Tools → Red Hat Certification** in the left panel.
4. Click the **Hosts** tab, and then click the host under test on which you want to run the tests.
5. Click **Provision**.
   A dialog box appears.
   a. Click **Upload,** and then select the new test plan .xml file. Then, click **Next**. A successful upload message is displayed.
      Optionally, if you want to reuse the previously uploaded test plan, then select it again to reupload.

NOTE

During the certification process, if you receive a redesigned test plan for the ongoing product certification, then you can upload it following the previous step. However, you must run **rhcert-clean all** in the Terminal tab before proceeding.

b. In the **Role** field, select **Host under test** and click **Submit**.

c. By default, the file is uploaded to path, **/var/rhcert/plans/<testplanfile.xml>**.

6.6. USING THE TEST PLAN TO PREPARE THE TEST SERVER FOR TESTING

Running the Provision Host command enables and starts the rhcertd service, which configures services specified in the test suite on the test server, such as iperf for network testing, and an nfs mount point used in kdump testing.

Prerequisites

- You have downloaded the test plan provided by Red Hat .

Procedure

1. Enter **http://*<Cockpit_system_IP>*:9090/** in your browser address bar to launch the Cockpit web application.
2. Enter the username and password, and then click **Login**.
3. Select **Tools →Red Hat Certification** in the left panel.
4. Click the **Hosts** tab, and then click the host under test on which you want to run the tests.
5. Click **Provision**.
   A dialog box appears.

   a. Click **Upload,** and then select the new test plan .xml file. Then, click **Next**. A successful upload message is displayed.
   Optionally, if you want to reuse the previously uploaded test plan, then select it again to reupload.

NOTE

During the certification process, if you receive a redesigned test plan for the ongoing product certification, then you can upload it following the previous step. However, you must run **rhcert-clean all** in the Terminal tab before proceeding.

   b. In the **Role** field, select **Test server** and click **Submit**. By default, the file is uploaded to the **/var/rhcert/plans/<testplanfile.xml>** path.

6.7. RUNNING THE CERTIFICATION TESTS USING COCKPIT

Prerequisites

- You have prepared the host under test .
- You have prepared the test server.

Procedure

1. Enter **http://<*Cockpit_system_IP*>:9090/** in your browser address bar to launch the Cockpit web application.
2. Enter the username and password, and click **Login**.
3. Select **Tools → Red Hat Certification** in the left panel.
4. Click the **Hosts** tab and click on the host on which you want to run the tests.
5. Click the **Terminal** tab and select **Run.**
   A list of recommended tests based on the test plan uploaded displays. The final test plan to run is a list of common tests taken from both the test plan provided by Red Hat and tests generated on discovering the system requirements.
6. When prompted, choose whether to run each test by typing **yes** or **no**.
   You can also run particular tests from the list by typing **select**.

6.8. REVIEWING AND DOWNLOADING THE TEST RESULTS FILE

Procedure

1. Enter **http://<*Cockpit_system_IP*>:9090/** in your browser address bar to launch the Cockpit web application.
2. Enter the username and password, and then click **Login**.
3. Select **Tools → Red Hat Certification** in the left panel.
4. Click the **Result Files** tab to view the test results generated.
   a. Optional: Click **Preview** to view the results of each test.
   b. Click **Download** beside the result files. By default, the result file is saved as **/var/rhcert/save/hostname-date-time.xml**.

6.9. SUBMITTING THE TEST RESULTS FROM COCKPIT TO THE RED HAT CERTIFICATION PORTAL

Procedure

1. Enter http://<Cockpit_system_IP>:9090/ in your browser's address bar to launch the Cockpit web application.
2. Enter the username and password, and then click **Login**.
3. Select **Tools → Red Hat Certification** in the left panel.

4. Click the **Result Files** tab and select the case number from the displayed list.

    a. For the authorized users click **Submit**. A message displays confirming the successful upload of the test result file.

    b. For non-authorized users see, Uploading the results file of the executed test plan to Red Hat Certification portal.

The test result file of the executed test plan will be uploaded to the Red Hat Certification portal.

6.10. UPLOADING THE TEST RESULTS FILE TO RED HAT CERTIFICATION TOOL

Use the Red Hat Certification Tool to submit the test results file of the executed test plan to the Red Hat Certification team.

Prerequisites

- You have downloaded the test results file from Cockpit or HUT.

Procedure

1. Log in to Red Hat Certification Tool.
2. On the homepage, enter the product case number in the search bar.
   Select the case number from the list that is displayed.
3. On the **Summary** tab, under the Files section, click **Upload**.

Next steps

Red Hat reviews the results file you submitted and suggest the next steps. For more information, visit Red Hat Certification Tool.

CHAPTER 7. CONFIGURING THE SYSTEMS AND RUNNING TESTS USING RHCERT CLI TOOL

To complete the certification process using CLI, you must prepare the host under test (HUT) and test server, run the tests, and retrieve the test results.

7.1. USING THE TEST PLAN TO PREPARE THE HOST UNDER TEST FOR TESTING

Running the provision command performs a number of operations, such as setting up passwordless SSH communication with the test server, installing the required packages on your system based on the certification type, and creating a final test plan to run, which is a list of common tests taken from both the test plan provided by Red Hat and tests generated on discovering the system requirements.

For instance, required hardware packages will be installed if the test plan is designed for certifying a hardware product.

Prerequisites

- You have the hostname or the IP address of the test server.

Procedure

1. Run the provision command in either way. The test plan will automatically get downloaded to your system.
   - If you have already downloaded the test plan:

     ```
     # rhcert-provision <path_to_test_plan_document>
     ```

     Replace <path_to_test_plan_document> with the test plan file saved on your system.

     Follow the on-screen instructions.
   - If you have not downloaded the test plan:

     ```
     # rhcert-provision
     ```

     Follow the on-screen instructions and enter your **Certification ID** when prompted.
2. When prompted, provide the hostname or the IP address of the test server to set up passwordless SSH. You are prompted only the first time you add a new system.

7.2. USING THE TEST PLAN TO PREPARE THE TEST SERVER FOR TESTING

Running the Provision command enables and starts the **rhcertd** service, which configures services specified in the test suite on the test server, such as iperf for network testing, and an nfs mount point used in kdump testing.

Prerequisites

- You have the hostname or IP address of the host under test.

Procedure

1. Run the provision command by defining the role, “test server”, to the system you are adding. This is required only for provisioning the test server.

```
# rhcert-provision --role test-server <path_to_test_plan_document>
```

   Replace <path_to_test_plan_document> with the test plan file saved on your system.

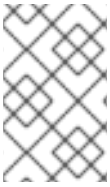
7.3. RUNNING THE CERTIFICATION TESTS USING CLI

Procedure

1. Run the following command:

```
# rhcert-run
```

2. When prompted, choose whether to run each test by typing **yes** or **no**.
   You can also run particular tests from the list by typing **select**.



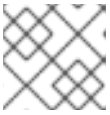
NOTE

After a test reboot, **rhcert** is running in the background to verify the image. Use **tail -f /*var*/log/rhcert/RedHatCertDaemon.log** to see the current progress and status of the verification.

7.4. SUBMITTING THE TEST RESULTS FILE

Procedure

1. Log in to authenticate your device.



NOTE

Logging in is mandatory to submit the test results file.

```
# rhcert-cli login
```

   a. Open the generated URL in a new browser window or tab.

   b. Enter the login and password and click **Log in**.

   c. Click **Grant access**.
      Device log in successful message displays.

   d. Return to the terminal and enter **yes** to the **Please confirm once you grant access**prompt.

2. Submit the result file.

```
# rhcert-submit
```

   When prompted, enter your Certification ID.

CHAPTER 8. CERTIFICATION WORKFLOW

8.1. ADDING CERTIFICATIONS TO PREVIOUSLY CERTIFIED HARDWARE

Use this process to create a new certification request for a system or component that has already completed a hardware certification process for an earlier RHEL version, or for a system or component that is being certified at the moment.

Procedure

1. Log in to the Red Hat Certification portal.
2. Click **New Certification**.
3. Select the Red Hat product, version, and platform for certification. Then, click **Next**.
4. Select vendor, make, and name of an already certified product from the dropdown lists. Then, click **Next**.

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

8.2. CHANGING FEATURES OR HARDWARE IN AN EXISTING CERTIFICATION

Apply for a supplemental certification to add hardware or features to an existing certification.

You can request a supplemental certification for features that were not certified before, for example, because they were not tested, or because their tests failed. After the additional features are certified, Red Hat will add the features to the certification catalog.

Procedure

1. Log in to the Red Hat Certification portal.
2. Click the existing hardware certification.
3. Click **Product**.
4. Click **Product Details**.
5. In the **Attachment** field, click **Browse** to attach the specification file for the new supplemental component. Select **is this a specification**
   The Red Hat certification team will add the supplemental components to the hardware certification.
6. Create the supplemental certification. You do not need to wait for the supplemental components to show in the existing hardware certification.
   a. Go to the hardware certification page.
   b. Click the **Certification** section.

c. Click the **Related Certification** tab, and go to the **Supplemental Certification** section.

d. Click **New Certification** to create a new supplemental certification.
The Red Hat certification team will add the test plan to the supplemental certification.

7. Run the certification tests. You do not need to wait for the new test plan.

8.3. CREATING A SYSTEM PASS-THROUGH CERTIFICATE USING EXISTING SPECIFICATION FILE

A system pass-through certification creates a copy of a certified system and lists 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 another.

Procedure

1. Log in to the Red Hat Certification portal.
2. Click the existing hardware certification.
3. Click **Related Certification**.
4. Click **Add Related Certification**, and select **Pass-through**.
5. Choose the appropriate product:
   - If the product has already been created, select it.
   - If the product is not in the list, create it as a new product.
6. Click **New Certification** to create the new pass-through certification.

The Red Hat certification team will review the hardware specification and publish the new system .certification. After the new certification is published, partners can refer to it as a pass-through certification.

8.3.1. Copying an existing system certification to a new entry

Procedure

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.

8.3.2. Creating a system pass-through certificate using existing specification file

Procedure

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 choose the pass through specification file that has been created.

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

8.4. CREATING AND PUBLISHING A COMPONENT PASS-THROUGH 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 others.

Procedure

1. Create a system certification. See Opening a new certification case by using the Red Hat Certification portal.
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.

8.4.1. Copying an existing component certification to a new entry

Procedure

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.

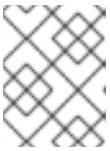
If the original component specifications and the pass-through component specifications are identical then, no additional testing will be required. If there are differences found, the Red Hat certification team will discuss with you what should be done to account for them.

8.5. ADDING MISSING DATA TO THE PRODUCT CERTIFICATION

To ensure accurate and complete certification information, follow this streamlined process for adding missing attributes before publishing the certification.

Procedure

1. Log in to the Red Hat Certification portal.
2. Click the existing hardware certification.
3. In the **Certification Status** section, click the question mark icon. A **Completion Requirements** notification banner displays the information about the missing attributes.
4. Click one of the missing attributes, and you will be redirected to the **Properties** tab of that certification.
5. Optional: Click the product under the **Partner Product** section and navigate to the **Properties** tab.
6. On the **Properties** tab, enter the missing details such as **Detail Description**, **Short Description**, **Partner Product Logo** or **Product Logo**, and **System Types**.



NOTE

The system type option is applicable only if you have selected the product category as **System**.

7. From the **System Types** list select one or many system types.
8. Add marketing URLs for different system types in the **Enter url** field.
9. Click **Update**.

Verification

The question mark icon is no longer visible if all the required data is present or updated.

After the certification is complete and published, the updated product data, along with the system types, will be available on the Red Hat Ecosystem Catalog.

NOTE

All the fields marked with an asterisk * are required and must be completed before publishing the certification.

8.6. CERTIFYING 64K KERNEL

The 64k page size kernel is a useful option for large datasets on ARM platforms. It is suitable for memory-intensive workloads as it has significant gains in overall system performance, especially in large databases, HPC, and high network performance.

From RHEL 9.2 onwards, ARM architecture uses the 64k page size kernel as optional and the 4k kernel as default. To certify the 64k page size kernel, you need to first complete RHEL 9 certification using the default 4k kernel and then you can conduct a second certification with the 64k kernel. After successful completion of the second certification, a Knowledgebase article will be attached to the 4k size kernel certification indicating support for the 64k page size with instructions on how to use the 64k kernel.

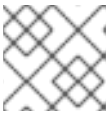
NOTE

You must create a supplemental certification to certify the 64k kernel.

8.7. DOWNLOADING GUEST IMAGES DURING TEST EXECUTION

Procedure

1. Verify if the guest images are available locally on the system. If yes, test execution will start.
2. If the guest images are not available locally, the test will try downloading them from the pre-configured test server.
3. If both local availability and the test server download fail, the test will establish a connection with the CWE API to obtain a pre-signed S3 URL for AWS. The guest image will then be downloaded from AWS by using the provided URL.
4. If the download from AWS also encounters an issue, the test will use CWE API to directly stream and download the guest images.
5. If all previous attempts to acquire the guest images are unsuccessful, the entire test is marked as FAIL.



NOTE

The above procedure is applicable for rhcert version 8.66 and later.

If the FV image download fails during the test run, follow these steps:

a. Download the files from the Red Hat Certification portal.

b. After downloading the files, move them to **/var/lib/libvirt/images** directory on the host under test.

c. To manually extract the files use the command **tar xmvfj <tarred file name>**.

d. After the file is extracted, rename it by using the command **mv <extracted file> <image file name>**. For example - **mv hwcertData-20211116.img hwcertData.img**.
Refer to the following table for the file names:

Tarred file name	Image file name

hwcertData.img.tar.bz2	hwcertData.img
hwcert-x86_64.img.tar.bz2	hwcert-x86_64.img
rhel-kvm-rt-image.qcow2.tar.bz2	rhel-kvm-rt-image.qcow2

CHAPTER 9. LAYERED PRODUCT CERTIFICATIONS

9.1. CERTIFYING LAYERED PRODUCTS

Layered product certifications are additional certifications awarded to the systems already certified for RHEL.

You can create layered product certifications in two ways:

1. Section 9.1.1, "Generating a layered certification automatically"
2. Section 9.1.2, "Creating a layered certification manually"

9.1.1. Generating a layered certification automatically

A layered certification can be generated automatically only if the status of the hardware certification is **Certified**. The Red Hat certification team makes the hardware certification **Public** for it to be listed on the Red Hat Certification portal.

Procedure

Perform the following steps to generate a layered certification automatically:

1. Log in to the Red Hat Certification portal.
2. Click the hardware certification that has to be certified and made public.
3. Click the **Dialog** tab.
4. In the **New Comment** text box, enter the comment to the Red Hat certification team to certify and make the certification public.
5. Click the **Add Comment** button.

After you add a comment for the requested hardware certification, the Red Hat certification team certifies and makes the certificate public. You will receive an email once the certification is certified and public.

You can see the new auto-created layered certification on the Red Hat Certification portal. If not, click the Refresh button, and the new certification should be downloaded.

9.1.2. Creating a layered certification manually

A layered certification can be created manually for the hardware certification if they have a classification as regular.

NOTE

- A layered certification is not supported for all the regular hardware certifications.
- Base certification should be published before publishing layered certification.

Procedure

Perform the following steps to create a layered certification manually:

1. On the Red Hat Certification portal, click the regular hardware certification certified by the Red Hat certification team.
2. Click the **Related Certification** tab.
3. Click **Add Related Certification**, and select **Layered** from the **Certification Types**.
4. Select the **Certification Type** from the **Red Hat Certification** and the **Product Version** from the **Red Hat Product Version** drop-down and click **Next**.
5. Review the certification information and click **Open**.

A new certification gets created in the Red Hat Certification portal, and you are redirected to the newly created certification page.

9.2. RED HAT ENTERPRISE LINUX FOR REAL-TIME

Supported RHEL versions	Supported RHEL for Real Time versions	Prerequisite certifications
7, 8, and 9	7, 8, and 9	RHEL

Apply for the Red Hat Enterprise Linux for Real Time certification after your system has been awarded the prerequisite certifications detailed in the previous table.

The Red Hat Hardware Certification test suite includes the necessary test to obtain a Real Time certification.

9.2.1. Additional resources

- Red Hat Enterprise Linux for Real Time Life Cycle

9.3. RED HAT VIRTUALIZATION

Supported RHEL versions	Supported RHEL for Real Time versions	Prerequisite certifications
7 and 8	7 and 8	RHEL

Red Hat Virtualization relies on and is co-engineered with RHEL. As a result of this common base, if your system passes the basic virtualization tests during the Red Hat Enterprise Linux certification, you do not need to run additional tests to receive the Red Hat Virtualization certification. If your system also passes the advanced virtualization tests, your Red Hat Virtualization certification will automatically also include the advanced features.

Red Hat will open the Red Hat Virtualization certification on your behalf automatically for all 64-bit Intel and AMD, and all IBM Power little endian server certifications submitted for RHEL.

If you want but did not automatically receive the certification, create a new layered certification request for the Red Hat Virtualization product.

9.3.1. Additional resources

- Red Hat Virtualization
- Red Hat Virtualization Life Cycle

9.4. RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM COMPUTE

Supported RHEL versions	Supported RHEL for Real Time versions	Prerequisite certifications
8 and 9	Consult the supportability matrix	RHEL (basic virtualization tests must have passed)

Red Hat OpenStack Platform relies on and is co-engineered with RHEL. As a result of this common base, you do not need to run additional tests to receive the Red Hat OpenStack Platform Compute certification if your system meets prerequisite certifications detailed in the previous table. You also do not need to open a new certification.

However, for Red Hat OpenStack Platform RHEL 8, layered product certification specifically applies to POWER systems that are not LPAR (logical partition) based. To qualify for layered product certification on RHEL 8, the system must function as a bare metal hypervisor during testing.

For RHEL 9, Red Hat doesn't support KVM on POWER systems. Hence there is no creation of virtualization-based layered product certifications.

For other architectures or categories, or in any situation where you want but did not automatically receive the certification, create a new layered certification request for the Red Hat OpenStack Platform Compute product.

Red Hat encourages partners certifying systems with baseboard management controllers (BMC) to apply for the Red Hat OpenStack Platform Bare Metal certification as well.

Additional resources

- Red Hat OpenStack Platform
- Red Hat OpenStack Platform Life Cycle
- Red Hat OpenStack Platform Hardware Bare Metal Certification Policy Guide
- Red Hat Bare Metal Hardware Certification Workflow Guide .

9.5. RED HAT OPENSTACK PLATFORM FOR REAL-TIME APPLICATIONS

Supported RHEL versions	Supported RHEL for Real Time versions	Prerequisite certifications

Supported RHEL versions	Supported RHEL for Real Time versions	Prerequisite certifications
8	Consult the supportability matrix	• RHEL (basic virtualization tests must have passed) • Red Hat Enterprise Linux for Real Time • Red Hat OpenStack Platform Compute

Apply for the Red Hat OpenStack Platform for Real-Time Applications certification after has been awarded the prerequisite certifications detailed in the previous table. The Red Hat Hardware Certification test suite includes the necessary test to obtain a Real Time certification.

9.6. RED HAT OPENSHIFT CONTAINER PLATFORM

Supported RHEL versions	Supported RHEL for Real Time versions	Prerequisite certifications
9.2, 9.3, and 9.4 (RHOCP 4.13, 4.14, or 4.15)	Consult the supportability matrix	RHEL (basic virtualization tests must have passed)
8 (RHOCP 4.12)	Consult the supportability matrix	RHEL (basic virtualization tests must have passed)

Red Hat OpenShift Container Platform relies on and is co-engineered with RHEL. As a result of this common base, you do not need to run additional tests to receive the Red Hat OpenShift Container Platform certification if your system meets prerequisite certifications detailed in the previous table.

However, for Red Hat OpenShift Container Platform, supported versions based on RHEL 8 (currently v4.11 and v4.12), layered product certification specifically applies to POWER systems that are not LPAR (logical partition) based. To qualify for layered product certification on RHEL 8, the system must function as a bare metal hypervisor during testing.

For RHEL 9, Red Hat doesn't support KVM on POWER systems. Hence there is no creation of virtualization-based layered product certifications.

For other architectures or categories, or in any situation where you want but did not automatically receive the certification, create a new layered certification request for the Red Hat OpenShift Container Platform product.

Apply for the Red Hat OpenShift Platform Bare Metal certification to add IPI and assisted installer capabilities to the RHOCP entry in the catalog.

Additional resources

- Red Hat OpenShift Container Platform

- Red Hat OpenShift Life Cycle
- Red Hat OpenStack Platform Hardware Bare Metal Certification Policy Guide
- Red Hat Bare Metal Hardware Certification Workflow Guide .

CHAPTER 10. 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.

10.1. 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

10.2. 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 the appropriate **Leverage** field of their system certification requests that contain this hardware.

10.3. GENERATING TEST RESULT ID FOR LEVERAGING FROM SYSTEM CERTIFICATION

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

Procedure

1. Create the source hardware product and certification from Red Hat Certification web user interface.
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**.

10.4. LEVERAGING FROM EXISTING COMPONENT

If you want to create a new certification using the same components, you can leverage that component in two ways:

Procedure

1. **Leveraging by copying Result ID**
   See Generating test result ID for leveraging from system certification .

   The Red Hat Certification team approves the components to leverage mentioned in the specification file.

   a. 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.

   b. If the ID is successfully copied, you will get a message **"Copied Leverage Information from System Certification<the_component_name>".**

   c. From the **Red Hat Certification** web user interface, click the hardware cert on which you want to add the leverage component.

   d. 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.
   If the leverage ID is successfully applied you will get a message **"Leverage of System Test successfully applied".**

2. **Leveraging using Result ID or Certification ID**

   a. Click on the **Certification** section. In the **Progress** tab, go to the **Test Result ID** column and click on the **Leverage Result**.

   b. On the **Leverage Result** window choose **Result ID** or **Certification ID** from **Leverage Using** drop-down.

   NOTE

   Use **Result ID** for leveraging using test result ID and choose **Certification ID** for leveraging using the pass-through certificate.

   c. Enter **Leverage ID** and click **Submit**.
   You will get a success message if Leveraging is successful, else a failure message is displayed if submitted ID has an error.

CHAPTER 11. REVIEWING TEST RESULTS AND COMPLETING CERTIFICATIONS

11.1. 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.

11.2. COMPLETING CERTIFICATIONS

A certification is complete after Red Hat confirms that all the items in the test plan have passed. At this point, you can choose whether to close and publish the certification or to close and leave the certification unpublished.

Supplemental certifications always remain unpublished. System and component certifications can be left unpublished if you do not want to advertise the certification status or the existence of the system or component.

The system information and the discussions between you and the Red Hat review team will not be visible to the public in the published certification.

If these publication options do not meet your requirements, submit a request for an exception while the certification is open, or a case if it is already closed.

APPENDIX A. TESTS

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.

RHEL version supported

This section lists the versions of RHEL that the test is supported 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.

You can choose either way to run the test:

- Follow Running the certification tests using CLI to run the test. Select the appropriate test name from the displayed list using the command:

```
rhcert-run
```

- In case of hardware detection issues or other hardware-related problems during planning, follow Manually adding and running the tests. Run the **rhcert-cli** command by specifying the desired test name.

```
rhcert-cli run --test=<test name>
```

Run Time

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

A.1. ACPI KEYS

What the test covers

The **ACPI keys** test captures a variety of input events from the system integrated keyboard.

RHEL version supported

- RHEL 8.6 and later

- RHEL 9

What the test does

The test captures the following:

- ACPI-related signals such as power, suspend, and sleep.
- Key presses that send signals associated with global keyboard shortcuts such as *<Meta+E>*, which opens the file browser.

Executing the test

The test is interactive. Run the following command and then select the appropriate **ACPI keys** test name from the list that displays.

```
rhcert-run
```

This test requires capturing all input events. During the test, press all the non-standard and multimedia keys on the device.

Press the **Escape key** at any time to end the test and to see a list of keys. The test is successful if all the keys that you tested appear in the list.

Run time

The test takes less than 5 minutes to finish. Any other mandatory or selected tests will add to the overall run time.

A.2. 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**.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 the **Settings** application click the **Sound** option.
2. Click on the **Output** tab, select the sound card you want to test, and adjust the **Output volume** to an appropriate level.
3. Click **Test Speakers**.
4. In the **Speaker Testing** pop-up window, click the **Test** buttons to generate sounds.

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). Verify the 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 notebook. Again, we don't require you to use a specific input jack; provided that one works, the test will pass.

1. In the **Settings** application click the **Sound** option.
2. Click on the **Input** tab, select the appropriate input device, and adjust the **Input volume** to 100%.
3. Speak into, tap, or otherwise activate the input device, and watch the **Input level** graphic. If you see it moving, the input device is set up properly. If it does not move, try another input selection or microphone port to plug the input device 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.

Run the following command and then select the appropriate **Audio** test name from the list that displays.

```
rhcert-run
```

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 **supportable** test will add about a minute to the overall run time.

A.3. BACKLIGHT

What the test covers

The **backlight** test runs when it detects an attached display on the system and support of software backlight control is available.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test ensures that the backlight control is functioning as expected by adjusting the display brightness of the attached display to the minimum and then maximum values.

Preparing for the test

- Ensure that the host under test is running RHEL 8.0 or later.
- Ensure that the system has backlight support and the display is connected to the system.

Executing the test

The test is interactive. Run the following command and then select the appropriate **backlight** test name from the list that displays.

```
rhcert-run
```

The display brightness will change from the maximum to minimum and back. The test will pass after you confirm that the display brightness is changing as expected.

Run time

This test takes less than a minute to run. Any other mandatory or selected tests will add to the overall run time.

A.4. BATTERY

What the test covers

The **battery** test is valid and can only be run on systems with built-in batteries. The test is not supported on external batteries that do not provide primary, internal power to the system, such as UPS or BIOS batteries. The test is scheduled when the hardware detection routines find the following string in the **udev** database:

```
POWER_SUPPLY_TYPE=Battery
```

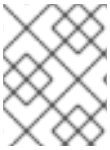
RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The test detects if the battery is connected, the AC adapter is plugged into the system, and the charging and discharging status of the battery.

Preparing for the test



NOTE

Do not perform the test when the battery is at 100% charge. Discharge the battery to a lower level before executing the test to avoid potential test failures.

Executing the test

The test is interactive. Run the following command and then select the appropriate **battery** test name from the list that displays.

```
rhcert-run
```

The test detects the 10 mWh battery charge and discharge, displays the current capacity, and charging status of the battery. Follow the on-screen instructions to unplug and plug in the AC adapter when prompted.

Run time

The execution time of the test depends on the charging and discharging speed of the battery. Since this test is run on a laptop, the required **supportable** test will run along with its **suspend** test. Overall, it takes 7-10 minutes.

A.5. BLUETOOTH

What the test covers

The **bluetooth** test is supported on systems with a bluetooth v3, v4, or v5 controller.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test uses **rfkill** command to check the availability of bluetooth controller in the system. It then runs **hciconfig** or **btmgmt** command to get the bluetooth controller version. Afterward, the test verifies if the

controller can scan, discover, pair with, select and trust another selected and aligned bluetooth v3, v4, or v5 device by using the **bluetoothctl** command tool. If the HUT has multiple bluetooth controllers, the bluetooth test is planned automatically for each bluetooth controller.

Preparing for the test

Before you begin the test, ensure to:

- Have a device that supports the same or later versions of Bluetooth as the controller.
- Enable Bluetooth on both the HUT and the pairing device.
- Pair the devices manually by using the settings application to confirm connectivity.
- Unpair the devices.

Executing the test

The test is interactive. Run the following command and then select the appropriate **bluetooth** test name from the list that displays.

```
rhcert-run
```

Then, select the device for which you want to test the bluetooth functionality.

Run time

The test takes around 5 minutes to complete. However, the time can vary depending on the bluetooth network connectivity.

A.6. BLURAY

What the test covers

The **Bluray** test runs on the following media and related drive types:

- Read-only media and drives (BD-ROM)
- Write-once media and drives (BD-R)
- Rewritable media and drives (BD-RE)

Based on information from the **udev** command, the test suite determines which optical drive tests (Blu-ray, DVD, or CD-ROM) to schedule and the type of media to test (read-only, writable, and rewritable). For example, the test suite will plan the following tests for a Blu-ray drive with rewriting capabilities that can also read DVD and CD-ROM discs:

- A rewrite (erase, write, and read) test for Blu-ray media
- A read test for DVD media
- A read test for CD-ROM media

You only need to run the Bluray test once for a given drive.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The test performs the following tasks depending on the capabilities of the drive:

- **Read-only drives** - First, it reads data from a disc and copies it to the hard disk. Then, it compares the data on the disc to the copy on the hard disk. If all file checksums match, the test passes.
- **Drives with writing capabilities** - First, it reads data from the hard disk and writes it to a writable blank disc. Then, it compares the data on the hard disk to the copy on the disc. If all file checksums match, the test passes.
- **Drives with rewriting capabilities** - First, it erases all information from the rewritable disc. Then, it reads data from the hard disk and writes it to the rewritable disc. Finally, it compares the data on the hard disk to the copy on the disc. If the erasing operation is successful and all file checksums match, the test passes.

Executing the test

The test is interactive. Run the following command and then select the appropriate **Bluray** test name from the list that displays.

```
rhcert-run
```

Follow the instructions on screen to insert the appropriate media and to close the drive's tray if appropriate.

Run time

The run time for the Bluray test depends on the speed of the media and the drive. For a 2x 25G BD-RE disc, the test finishes in approximately 14 minutes.

A.7. CD ROM

What the test covers

The **CD ROM** test runs on the following media and related drive types:

- Read-only media and drives (CD-ROM)
- Write-once media and drives (CD-R)
- Rewritable media and drives (CD-RW)

Based on information from the **udev** command, the test suite determines which optical drive tests (Blu-ray, DVD, or CD-ROM) to schedule and the type of media to test (read-only, writable, and rewritable). For example, the test suite will plan the following tests for a Blu-ray drive with rewriting capabilities that can also read DVD and CD-ROM discs:

- A rewrite (erase, write, and read) test for Blu-ray media

- A read test for DVD media
- A read test for CD-ROM media

You only need to run the CD ROM test once for a given drive.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The test performs the following tasks depending on the capabilities of the drive:

- **Read-only drives** - First, it reads data from a disc and copies it to the hard disk. Then, it compares the data on the disc to the copy on the hard disk. If all file checksums match, the test passes.
- **Drives with writing capabilities** - First, it reads data from the hard disk and writes it to a writable blank disc. Then, it compares the data on the hard disk to the copy on the disc. If all file checksums match, the test passes.
- **Drives with rewriting capabilities** - First, it erases all information from the rewritable disc. Then, it reads data from the hard disk and writes it to the rewritable disc. Finally, it compares the data on the hard disk to the copy on the disc. If the erasing operation is successful and all file checksums match, the test passes.

Executing the test

The test is interactive. Run the following command and then select the appropriate **CD ROM** test name from the list that displays.

```
rhcert-run
```

Follow the instructions on screen to insert the appropriate media and to close the drive's tray if appropriate.

Run time

The run time for the CD ROM test is dependent on the speed of the media and drive. For a 12x 714MB CD-RW disc, the test finishes in approximately 7 minutes.

A.8. CORE

What the test covers

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

RHEL version supported

- RHEL 7

- RHEL 8
- RHEL 9

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 **getimeofday()** 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. Run the following command and then select the appropriate **Core** test name from the list that displays.

```
rhcert-run
```

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 **supportable** 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 **supportable** test that runs in the guest. The required **supportable** test on the bare-metal system will add about a minute to the overall run time.

A.9. CPU SCALING

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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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/cpu*X*/topology/physical_package_id**, and run the test in parallel for all the logical CPUs in a particular package.

The test runs the turbostat command first to gather the processor statistics. On supported architectures, turbostat checks if the advance statistics columns are visible in the turbostat output file, but returns a warning if the file does not contain the columns. The test then attempts to execute the cstate subtest and if it fails, executes pstate subtest.

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/cpu*X*/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 2x10^12 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. Run the following command and then select the appropriate **CPU scaling** test name from the list that displays.

```
rhcert-run
```

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 **supportable** test will add about a minute to the overall run time.

A.10. DVD

What the test covers

The **DVD** test runs on the following media and related drive types:

- Read-only media and drives (DVD-ROM)
- Write-once media and drives (DVD+R and DVD-R)
- Rewritable media and drives (DVD+RW and DVD-RW)

Based on information from the **udev** command, the test suite determines which optical drive tests (Blu-ray, DVD, or CD-ROM) to schedule and the type of media to test (read-only, writable, and rewritable). For example, the test suite will plan the following tests for a Blu-ray drive with rewriting capabilities that can also read DVD and CD-ROM discs:

- A rewrite (erase, write, and read) test for Blu-ray media
- A read test for DVD media
- A read test for CD-ROM media

If your drives support both the DVD-RW and DVD+RW formats, you can use either type of disc during the test. You do not need to test both formats. Moreover, you only need to run the DVD test once for a given drive.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The test performs the following tasks depending on the capabilities of the drive:

- **Read-only drives** - First, it reads data from a disc and copies it to the hard disk. Then, it compares the data on the disc to the copy on the hard disk. If all file checksums match, the test passes.

- **Drives with writing capabilities** - First, it reads data from the hard disk and writes it to a writable blank disc. Then, it compares the data on the hard disk to the copy on the disc. If all file checksums match, the test passes.
- **Drives with rewriting capabilities** - First, it erases all information from the rewritable disc. Then, it reads data from the hard disk and writes it to the rewritable disc. Finally, it compares the data on the hard disk to the copy on the disc. If the erasing operation is successful and all file checksums match, the test passes.

Executing the test

Run the following command and then select the appropriate **DVD** test name from the list that displays.

```
rhcert-run
```

Follow the instructions on screen to insert the appropriate media and to close the drive's tray if appropriate.

Run time

The run time for the DVD test is dependent on the speed of the media and drive. For a 4x 4.7GB DVD-RW disc, the test finishes in approximately 13 minutes.

A.11. 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.

NOTE

You must manually start the **httpd** service if you are using a RHEL 8 test server with a RHEL 7 host under test for running this test as the test suite does not start this service automatically.

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
```

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

A.12. 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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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. Run the following command and then select the appropriate **Expresscard** test name from the list that displays.

```
rhcert-run
```

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.13. FINGERPRINTREADER

What the test covers

The **fingerprintreader** test is planned if the system has a built-in or plugin fingerprint reader.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

This test verifies that a fingerprint reader can scan, enroll, and verify the enrolled fingerprints in the fingerprint manager.

Preparing for the test

Ensure that the fingerprint reader is connected to the system.

Executing the test

This test is interactive. Run the following command and then select the appropriate **fingerprintreader** test name from the list that displays.

```
rhcert-run
```

The test will start detecting the fingerprint reader and then prompt you to place and scan your right index finger several times on the fingerprint reader until the enrollment completes. For verification, you will be prompted to scan the finger again for matching it with the enrolled fingerprint.

Run time

The test takes around a couple of minutes to complete until the reader finishes scanning and shows the *enroll-complete* state.

A.14. FIRMWARE

What the test covers

The firmware test is supported to run on RHEL versions 8 and later for the x86_64 architecture systems using Unified Extensible Firmware Interface (UEFI) and the EFI System Resource Table (ESRT) for firmware management only.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test runs the following subtests:

- Security Check subtest: The subtest checks if the host under test follows the security best practices by validating if the system and device firmware meet HSI-1 level standards. The test uses the **fwupdagent security --force** command to check the HSI-1 security attributes and capture the output.
- Update Service subtest: The subtest verifies if the host under test can download and install the firmware updates through Linux Vendor Firmware Service (LVFS).

Success criteria

- The test passes only if all the HSI-1 attributes pass.
- The test passes if the system installs the LVFS update.

Preparing for the test

- Ensure that the host under test is running RHEL 8.0 or later.
- Ensure that the system is booted in UEFI mode and not legacy BIOS mode.

Executing the test

This test is noninteractive. Run the following command and then select the appropriate **firmware** test name from the list that displays.

```
rhcert-run
```

Run time

This test takes a minute to run. Any other mandatory or selected tests will add to the overall run time.

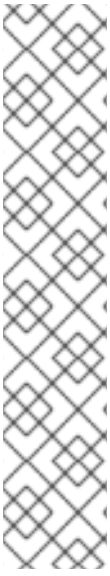
A.15. FV_CORE

The **fv_core** test is a wrapper that launches the FV guest and runs a **core** test on it.

Starting with RHEL 9.4, this test is supported to run on ARM systems.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9



NOTE

The first time you run any full-virtualization test, the test tool will need to obtain the FV guest files. The execution time of the test tool depends on the transfer speed of the FV guest files. For example,

- If FV guest files are located on the test server and you are using 1GbE or faster networking, it takes almost a minute or two to transfer approximately 300MB of guest files.
- If the files are retrieved from the CWE API, which occurs automatically when the guest files are not installed or found on the test server, the first runtime will depend on the transfer speed from the CWE API.

When the guest files are available on the Host Under Test (HUT), they will be utilized for all the later runs of fv_* tests.

Additional resources

- For more information about the test methodology and run times, see core.
- For more information about guest images, see Downloading guest images during test execution.

A.16. FV_CPU_PINNING

CPU pinning is a method for dedicating system resources to a particular process. For example, an application may be locked to a particular logical core to reduce task switching.

The virtualized (fv) CPU pinning method is similar except that pinning is done from a virtual CPU (vCPU) inside a KVM-based virtual machine to a physical core on the host machine.

Starting with RHEL 9.4, this test is supported to run on ARM systems.

RHEL version supported

- RHEL 8
- RHEL 9

What the test covers

The **fv_cpu_pinning** test validates that a vCPU of the guest virtual machine (VM) can be configured and pinned to a dedicated CPU of the host machine. This test is run on a host machine and is supported on RHEL 8 for feature qualification on RHEL 8 based RHV 4 releases.

What the test does

The **fv_cpu_pinning** test runs three subtests: Setup guest VM VCPU, Perform FV CPU Pinning, and verify FV CPU Pinning. The Setup guest VM VCPU subtest counts the number of logical cores of the host machine and isolates the last numbered core among those to dedicate it to the VM. The Perform FV CPU Pinning subtest further pins the vCPU of the VM to the CPU in the host machine. The test then verifies the pinning using commands, *virt vcpupin* and *vcpuinfo*, and checking the /proc directory information. Finally, the verify FV CPU Pinning uses the load test to verify if the guest VM vCPU workload is handled by the pinned CPU only.

Preparing for the test

There are no special requirements to run this test.

Executing the test

The fv_cpu_pinning test is non-interactive. Run the following command and then select the appropriate **fv_cpu_pinning** test name from the list that displays.

```
rhcert-run
```

Run time

The fv_cpu_pinning test takes around 5 minutes to complete. Any other mandatory or selected tests will add to the overall run ti

Additional resources

- For more information about guest images, see Downloading guest images during test execution.

A.17. FV_LIVE_MIGRATION

What the test covers

The **fv_live_migration** test checks the ability of a Host Under Test (HUT) to migrate a running virtual machine to a test server.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test performs multiple subtests to complete the migration of a running virtual machine from the HUT to test server. Successful completion of the test requires all of the subtests to pass. The test checks if the HUT meets the requirements for migration, configures a virtual machine, and starts it on the HUT. It then migrates the running virtual machine from the HUT to the test server. After migration, verifies the virtual machine is no longer running on the HUT and is running on the test server. Finally, the test migrates the running virtual machine from the test server back to the HUT and checks again that the virtual machine is running on the HUT and no longer running on the test server.

Preparing for the test

Ensure that test server and HUT are running Red Hat Enterprise Linux 8, and the **redhat-certification-hardware** package is installed on the test server and HUT.

Add the hostname in the respective */etc/hosts* file in both test server and HUT and make the hostname alias for the fully qualified name as shown below:

- *<IP address of HUT> <hostname of HUT>*
- *<IP address of test server> <hostname of test server>*

Executing the test

The test is non-interactive. Currently, this test can be planned and executed manually via CLI only.

On RHEL 8:

```
# rhcert-cli plan --add -t fv_live_migration
```

```
# rhcert-cli run -t fv_live_migration --server=<server name>
```

On RHEL 9:

```
# rhcert-cli plan --add -t fv_live_migration
```

```
# rhcert-cli run --test fv_live_migration --server=<server name>
```

Run time

The test takes around 5 minutes to complete. However, the time might decrease or increase if the test server and HUT belong to the same or a different lab or network respectively.

Additional resources

- For more information about guest images, see Downloading guest images during test execution.

A.18. FV_MEMORY

The **fv_memory** test is a wrapper that launches the FV guest and runs a **memory** test on it.

Starting with RHEL 9.4, this test is supported to run on ARM systems.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

NOTE

The first time you run any full-virtualization test, the test tool will need to obtain the FV guest files. The execution time of the test tool depends on the transfer speed of the FV guest files. For example,

- If FV guest files are located on the test server and you are using 1GbE or faster networking, it takes almost a minute or two to transfer approximately 300MB of guest files.
- If the files are retrieved from the CWE API, which occurs automatically when the guest files are not installed or found on the test server, the first runtime will depend on the transfer speed from the CWE API.

When the guest files are available on the Host Under Test (HUT), they will be utilized for all the later runs of fv_* tests.

Additional resources

- For more information about the test methodology and runtime, see memory.
- For more information about guest images, see Downloading guest images during test execution.

A.19. FV_PCIE_STORAGE_PASSTHROUGH

What the test covers

The **fv_pcie_storage_passthrough** test is used to verify that control over a PCIe-based storage device, such as SAS and SATA, in the host machine can be transferred to a virtual machine. The test is supported on Red Hat Enterprise Linux 8 and must be run on a host machine. This test is planned automatically if the host supports device passthrough and has IOMMU enabled.

Starting with RHEL 9.4, this test is supported to run on ARM systems.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test performs multiple subtests to attach a host machine's HBA device to a virtual machine and then run the storage tests inside the virtual machine. Successful completion of the test requires all of the subtests to pass. The test validates if the PCIe device connected to the host machine can be assigned to appear natively in the guest virtual machine, configures the guest virtual machine to use the passthrough PCIe device, and launches the virtual machine and ensures the device is functioning as expected inside it.

Preparing for the test

Ensure that the host machine supports device passthrough and has IOMMU enabled. To configure, see Configuring a Host for PCI Passthrough .

NOTE

Do not run the test on the storage devices with the root partition of the host machine.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **fv_pcie_storage_passthrough** test name from the list that displays.

```
rhcert-run
```

Run time

The test takes around 30 minutes to run.

Additional resources

- For more information about guest images, see Downloading guest images during test execution .

A.20. FV_USB_NETWORK_PASSTHROUGH

What the test covers

The **fv_usb_network_passthrough** test is used to verify that control over a USB-attached network device in the host machine can be transferred to a virtual machine. The test is supported on Red Hat Enterprise Linux version 8 and above and must be run on a host machine. This test is planned automatically if the host machine supports device passthrough and has IOMMU enabled.

Starting with RHEL 9.4, this test is supported to run on ARM systems.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test performs multiple subtests to attach a host machine's USB device to a virtual machine and then run the network tests inside the virtual machine. Successful completion of the test requires all of the subtests to pass. The test validates if the USB device connected to the host machine can be assigned to appear natively in the guest virtual machine, configures the guest virtual machine to use the passthrough USB device, and launches the virtual machine and ensures the device is functioning as expected inside it.

Preparing for the test

- Ensure that the USB device is plugged into the HUT that supports device passthrough and has IOMMU enabled. To configure, see Configuring a Host for PCI Passthrough .
- Ensure that the HUT has a minimum of two NIC and both networks are routable to the test server.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **fv_usb_network_passthrough** test name from the list that displays.

```
rhcert-run
```

NOTE

If the test fails due to network bandwidth issues, then you might have to increase the CPUs and RAM allocated to the virtual machine to achieve higher bandwidth.

Run time

The test takes around 90 minutes to run, but will vary in length depending on the size and speed of the USB device and connection.

Additional resources

- For more information about guest images, see Downloading guest images during test execution.

A.21. FV_USB_STORAGE_PASSTHROUGH

What the test covers

The **fv_usb_storage_passthrough** test is used to verify that control over a USB-attached storage device, in the host machine can be transferred to a virtual machine. The test is supported on Red Hat Enterprise Linux 8 and must be run on a host machine. This test is planned automatically if the host supports device passthrough and has IOMMU enabled.

Starting with RHEL 9.4, this test is supported to run on ARM systems.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test performs multiple sub tests to attach a host machine's USB device to a virtual machine and then run the storage tests inside the virtual machine. Successful completion of the test requires all of the subtests to pass. The test validates if the USB device connected to the host machine can be assigned to appear natively in the guest virtual machine, configures the guest virtual machine to use the passthrough USB device, and launches the virtual machine and ensures the device is functioning as expected inside it.

Preparing for the test

Ensure that the USB device is plugged into the host machine that supports device passthrough and has IOMMU enabled. To configure, see Configuring a Host for PCI Passthrough section in the *Red Hat Virtualization Administration Guide*.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **fv_usb_storage_passthrough** test name from the list that displays.

```
rhcert-run
```

Run time

The test takes around 90 minutes to run, but will vary in length depending on the size and speed of the USB device and connection.

Additional resources

- For more information about guest images, see Downloading guest images during test execution.

A.22. FV_PCIE_NETWORK_PASSTHROUGH

What the test covers

The **fv_pcie_network_passthrough** test is used to verify that control over a PCIe-based network device, such as NIC, LOMs, ALOMs, in the host machine can be transferred to a virtual machine. The test is supported on Red Hat Enterprise Linux version 8 and above, and must be run on a host machine. This test is planned automatically if the host machine supports device passthrough and has IOMMU enabled.

Starting with RHEL 9.4, this test is supported to run on ARM systems.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test performs multiple subtests to attach a host machine's network device to a virtual machine and then run the network tests inside the virtual machine. Successful completion of the test requires all of the subtests to pass. The test validates if the PCIe device connected to the host machine can be assigned to appear natively in the guest virtual machine, configures the guest virtual machine to use the passthrough PCIe device, and launches the virtual machine and ensures the device is functioning as expected inside it.

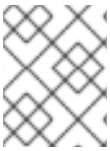
Preparing for the test

- Ensure that the Host Under Test (HUT) supports device passthrough and has IOMMU enabled. To configure, see Configuring a Host for PCI Passthrough section in the *Red Hat Virtualization Administration Guide*.
- Ensure that the HUT has a minimum of two NIC and both networks are routable to the test server.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **fv_pcie_network_passthrough** test name from the list that displays.

```
rhcert-run
```



NOTE

If the test fails due to network bandwidth issues, then you might have to increase the CPUs and RAM allocated to the virtual machine to achieve higher bandwidth.

Run time

The test takes around 30 minutes to run.

Additional resources

- For more information about guest images, see Downloading guest images during test execution.

A.23. INFINIBAND CONNECTION

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 HUT to the selected IP address of the test server.
2. Rping test
   Runs rping on test server and HUT using the selected test server IP address, then compares results to verify it ran to completion.
3. Rcopy test
   Runs rcopy on test server and HUT, sending a randomly generated file and comparing md5sums on test server and HUT to verify the 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 test server using device and port for another verification the device/port has been registered with the fabric.

8. ib_write_bw test
   Run ib_write_bw from the HUT to the selected IP address of the test server to test the InfiniBand write bandwidth and verify if it can reach the required bandwidth. The queue pair parameter has been adjusted during the bandwidth test to achieve a throughput closer to the line rate.

9. ib_read_bw test
   Run ib_read_bw from the HUT to the selected IP address of the test server to test the InfiniBand read bandwidth and verify if it can reach the required bandwidth. The queue pair parameter has been adjusted during the bandwidth test to achieve a throughput closer to the line rate.

10. ib_send_bw test
    Run ib_send_bw from the HUT to the selected IP address of the test server to test the InfiniBand send bandwidth and verify if it can reach the required bandwidth. The queue pair parameter has been adjusted during the bandwidth test to achieve a throughput closer to the line rate.

Preparing for the test

- Ensure that the test server and HUT are separate machines, on the same fabric(s).

Executing the test

This is an interactive test. Run the following command and then select the appropriate **infiniband connection** test name from the list that displays.

```
rhcert-run
```

You will be prompted with a dropdown to select an IP address (an IP address of test server) in which to perform the tests. Select an IP address corresponding to a device on the same fabric of the HUT device you are running the test for.

Table A.1. Manually adding and running the test

Rate Type	Command to manually add infiniband connection Test	Command to Manually run infiniband connection Test

Rate Type	Command to manually add infiniband connection Test	Command to Manually run infiniband connection Test
Infiniband_QDR	rhcert-cli plan --add --test Infiniband_QDR --device <devicename>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test Infiniband_QDR --server <test server IP addr>
Infiniband_FDR	rhcert-cli plan --add --test Infiniband_FDR --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test Infiniband_FDR --server <test server IP addr>
Infiniband_EDR	rhcert-cli plan --add --test Infiniband_EDR --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test Infiniband_EDR --server <test server IP addr>
Infiniband_HDR	rhcert-cli plan --add --test Infiniband_HDR --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test Infiniband_HDR --server <test server IP addr>
Infiniband_NDR	rhcert-cli plan --add --test Infiniband_NDR --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test Infiniband_NDR --server <test server IP addr>
Infiniband_Socket_Direct	rhcert-cli plan --add --test Infiniband_Socket_Direct	rhcert-cli run --test Infiniband_Socket_Direct --server <test server IP addr>

Replace **<device name>**, **<port number>**, **<net device>**, and **<test server IP addr>** with the appropriate value.

Run time

This test takes less than 10 minutes to run.

Additional resources

- For more information about InfiniBand and RDMA, see Understanding InfiniBand and RDMA technologies.

A.24. INTEL_SST

What the test covers

The **intel_sst** test is a CPU frequency scaling test that exercises Intel's Speed Select Technology (SST) feature. Use this feature to customize the per-core performance to match CPU to workload, and to allocate per performance. This enables you to boost the performance of targeted applications at runtime.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The **intel_sst test** runs on SST-enabled systems only and supports the following features:

- Speed Select Base Freq (SST-BF) - Allows specific cores to run higher base frequency (P1) by reducing the base frequencies (P1) of other cores.
- Frequency Prioritization (SST-CP) - Allows specific cores to clock higher by reducing the frequency of cores running lower-priority software.

The test checks if the above features are supported and configured on the system. Based on the result, it will execute only one subtest of the respective feature.

Preparing for the test

You must run this test on Intel chipset architectures only.

- To use the Intel® SST-BF functionality on a Red Hat Enterprise Linux (RHEL) based platform. *Prerequisites*:
  1. Enable the Intel® SST-BF feature in the BIOS
  2. Configure the kernel parameters - **intel_idle.max_cstate=1**
- To use the Intel® SST-CP functionality on a Red Hat Enterprise Linux (RHEL) based platform. *Prerequisites*:
  1. Enable the Intel® SST-CP feature in the BIOS
  2. Configure the kernel parameters - **intel_idle.max_cstate=1 intel_pstate=disable**

Executing the test

This test is non-interactive. Run the following command and then select the appropriate **intel_sst** test name from the list that displays.

```
rhcert-run
```

Run time

This test takes around 5 minutes to complete. Any other mandatory or selected tests will add to the overall run time.

A.25. IPXE

What the iPXE test covers

The iPXE test is an interactive test that runs on x86 Red Hat Enterprise Linux (RHEL) systems. The system should boot in UEFI boot mode.

If the **efi directory** exists the machine is running in the UEFI boot mode. Run the following command to determine if your machine is running in UEFI mode:

```
ls /sys/firmware/efi/
```

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

iPXE is the leading open source network boot firmware. It provides a full PXE implementation enhanced with additional features such as boot from HTTP, SAN, and Wireless Network. This test checks if the underlying NIC supports iPXE by using the HTTP boot.

While performing the iPXE the test server does not return any bootable image. The boot screen will display an error **could not boot**, this is an expected error message. The test server will boot with the next boot loader that is with the RHEL OS.

Preparing for the test

- Ensure that the host under test is in the UEFI boot mode. iPXE tests the interface that it finds first, thus, on the host under test, ensure the interface which needs to be tested is plugged in.
- Ensure that **httpd** service is not running on the test server while running this test, as this test uses port 80 to communicate with the test server.

Executing the test

1. Run the following command and then select the appropriate **iPXE** test name from the list that displays.

   ```
   rhcert-run
   ```

2. The ipxe test does not appear in the test plan, so you must use the following commands to plan and execute it manually, respectively.

```
# rhcert-cli plan --add --test iPXE
```

```
# rhcert-cli run --test iPXE
```

3. The test will first configure the Host Under Test (HUT) for iPXE test. It will save the MAC details of HUT, then it will create a new boot loader with ipxe binary and mark the boot loader as the next boot. After that, it will prompt for a reboot, press Yes to continue. The test server will display waiting for a response after it sends the reboot command.

4. The HUT will be rebooted to the new boot loader, which in turn loads the iPXE prompt and do a GET request to see if it is able to reach the test server. As it is just a GET requets the boot will fail and the system will fall back to the next boot loader i.e. RHEL OS.

5. The test server will continuously monitor the host under test to see if it has rebooted. After the reboot, the test will continue. The test will first revert the boot changes done for iPXE and then verify if iPXE boot was successful.

6. It will compare the MAC address received from the GET request of iPXE boot with the MAC already saved. If the MAC matches the iPXE test is successful.

Run time

The test takes less than 5 minutes to run. Any other mandatory or selected tests will add to the overall run time.

A.26. IWARP CONNECTION

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 HUT to the selected IP address of the test server.

2. Rping test - Runs rping on test server and HUT using the selected test server IP address, then compares results to verify it ran to completion.

3. Rcopy test - Runs rcopy on test server and HUT, sending a randomly generated file and comparing md5sums on test server and HUT to verify successful transfer.

4. Ethtool test - Runs the ethtool command passing in the detected net device of the roce device.

5. ib_write_bw test

Run ib_write_bw from the HUT to the selected IP address of the test server to test the IWrap write bandwidth and verify if it can reach the required bandwidth.

6. ib_read_bw test
   Run ib_read_bw from the HUT to the selected IP address of the test server to test the IWrap read bandwidth and verify if it can reach the required bandwidth.

7. ib_send_bw test
   Run ib_send_bw from the HUT to the selected IP address of the test server to test the IWrap send bandwidth and verify if it can reach the required bandwidth.

Preparing for the test

- Ensure that the test server and HUT are separate machines, on the same fabric(s).

Executing the test

This is an interactive test. Run the following command and then select the appropriate **iwrap connection** test name from the list that displays.

```
rhcert-run
```

You will be prompted with a dropdown to select an IP address (an IP address of test server) in which to perform the tests. Select an IP address corresponding to a device on the same fabric of the HUT device you are running the test for.

Table A.2. Manually adding and running the test

Speed Type	Command to manually add IWarpConnection Test	Command to Manually run IWarpConnection Test
10GigiWarp	`rhcert-cli plan --add --test 10GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device>`	`rhcert-cli run --test 10Gigiwarp --server <test server IP addr>`
20GigiWarp	`rhcert-cli plan --add --test 20GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device>`	`rhcert-cli run --test 20GigiWarp --server <test server IP addr>`

Speed Type	Command to manually add IWarpConnection Test	Command to Manually run IWarpConnection Test
25GigiWarp	rhcert-cli plan --add --test 25GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 25GigiWarp --server <test server IP addr>
40GigiWarp	rhcert-cli plan --add --test 40GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 40GigiWarp --server <test server IP addr>
50GigiWarp	rhcert-cli plan --add --test 50GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 50GigiWarp --server <test server IP addr>
100GigiWarp	rhcert-cli plan --add --test 100GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 100GigiWarp --server <test server IP addr>
200GigiWarp	rhcert-cli plan --add --test 200GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 200GigiWarp --server <test server IP addr>

Replace **<device name>**, **<port number>**, **<net device>**, and **<test server IP addr>** with the appropriate values.

Run time

This test takes less than 10 minutes to run.

Additional resources

- For more information about InfiniBand and RDMA, see Understanding InfiniBand and RDMA technologies.

A.27. KDUMP

What the test covers

The **kdump** test uses the **kdump** service to check that the system can capture a **vmcore** file after a crash, and that the captured file is valid.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

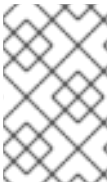
What the test does

The test includes the following subtests:

- **kdump with local**: Using the **kdump** service, this subtest performs the following tasks:
  - Crashes the host under test (HUT).
  - Writes a **vmcore** file to the local **/var/crash** directory.
  - Validates the **vmcore** file.
- **kdump with NFS**: Using the **kdump** service, this subtest performs the following tasks:
  - Mounts the **/var/rhcert/export** filesystem on the HUT's **/var/crash** directory. This filesystem is shared over NFS from the test server.
  - Crashes the HUT.
  - Writes a **vmcore** file to the **/var/crash** directory.
  - Validates the **vmcore** file.

Preparing for the test

- Ensure that the HUT is connected to the test server before running the test.
- Ensure that the **rhcertd** process is running on the test server. The certification test suite prepares the NFS filesystem automatically. If the suite cannot set up the environment, the test fails.



NOTE

You must manually start the **httpd** service if you are using a RHEL 8 test server with a RHEL 7 host under test for running this test as the test suite does not start this service automatically.

Executing the test

1. Log in to the HUT.
2. Run the **kdump** test:
   - To use the **rhcert-run** command, perform the following steps:
     i. Run the **rhcert-run** command:

     ```
     # rhcert-run
     ```

     ii. Select the **kdump** test.
     The test runs both subtests sequentially.
   - To use the **rhcert-cli** command, choose whether to run both subtests sequentially, or specify a subtest:
     - To run **both** subtests sequentially, use the following command:

     ```
     # rhcert-cli run -test=kdump –server=<test server's IP>
     ```

     - To run the **kdump with local** subtest only, use the following command:

     ```
     # rhcert-cli run -test=kdump -device=local
     ```

     - To run the **kdump with NFS** subtest only, use the following command:

     ```
     # rhcert-cli run -test=kdump -device=nfs –server=<test server's IP>
     ```

     Additionally, for the kdump with NFS test, execute the following command on the Test Server:

     ```
     # rhcertd start
     ```

3. Wait for the HUT to restart after the crash.
   The **kdump** service shows several messages while it saves the **vmcore** file to the **/var/crash** directory. After the **vmcore** file is saved, the HUT restarts.
4. Log in to the HUT after reboot, the rhcert suite will verify if the **vmcore** file exists, and if it is valid. If the file does not exist or is invalid, the test fails.

If you are running the subtests sequentially, the **kdump with NFS** subtest starts after the validation of the previous **vmcore** file has completed.

Run time

The run time of the **kdump** test varies according to factors such as the amount of RAM in the HUT, the disc speed of the test server and the HUT, the network connection speed to the test server, and the time taken to reboot the HUT.

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 four minutes, including the reboot. The same 2013-era workstation can complete an **NFS kdump** test in

about five minutes to a similarly equipped network test server. The **supportable** test will add about a minute to the overall run time.

A.28. 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"
```

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 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. Run the following command and then select the appropriate **lid** test name from the list that displays.

```
rhcert-run
```

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 **supportable** test for each run. The suspend test will add approximately 6 minutes to each test run, and **supportable** will add another minute.

A.29. 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 tests main memory only.

A **memory per CPU core check**has been added to the planning process to verify that the HUT meets the RHEL minimum requirement memory standards. It is a planning condition for several of the hardware certification tests, including the ones for memory, core, realtime, and all the full-virtualization tests.

If the **memory per CPU core check**does not pass, the above-mentioned tests will not be planned automatically. However, these tests can be planned manually via CLI.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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. Run the following command and then select the appropriate **memory** test name from the list that displays.

```
rhcert-run
```

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, AMD64 and Intel 64. The test will take longer on systems with more RAM. The required **supportable** 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 **supportable** test that runs in the guest. The required **supportable** 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.

Creating and Activating Swap for EC2 Partners can perform the following steps to create and activate swap for EC2

```
sudo dd if=/dev/zero of=/swapfile bs=1M count=8000
chmod 600 /swapfile
mkswap /swapfile
swapon /swapfile
swapon -s
edit file /etc/fstab and add the following line:
/swapfile swap swap defaults 0 0
write file and quit/exit
```

A.29.1. memory_HBM

What the memory_HBM tests cover

The **memory_HBM** tests are used to test system High Bandwidth Memory (HBM) on systems with it present. One of the three possible tests is planned based on the HBM operating mode. If the system HBM is not supported a regular **memory** test is planned instead.

RHEL version supported

- RHEL 8
- RHEL 9

What the tests do

The **memory_HBM** tests are memory tests specifically for systems with HBM present.

Preparing for the test

Ensure that the system HBM meets the requirements specified in the Policy Guide.

Executing the test

1. Run **rhcert-cli plan**. One of the memory_HBM tests will be planned if your HBM configuration meets the requirements and one of the following conditions:
   - **memory_HBM_only**: There is no DIMM installed in the system
   - **memory_HBM_cache**: HBM acts as cache to the DIMM
   - **memory_HBM_flat**: DIMM and HBM are available as a total amount of memory
2. To run the test, use the command **rhcert-cli run --test**. For example, **rhcert-cli run --test hwcert/memory_HBM_cache** runs the **memory_HBM_cache** test.
3. Follow the instructions of the test.

4. Use **rhcert-print** to check the result.
5. To save the result use **rhcert-save**.

A.30. NETWORK

What the test covers

The network test checks devices that transfer data over a TCP/IP network. The test can check multiple connection speeds and bandwidths of both wired and wireless devices based on the corresponding test designed for it, as listed in the following table:

Different tests under Network test

Ethernet test	Description
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.
200GigEthernet	The network test with added speed detection for 200 gigabit Ethernet connections.
Ethernet	If the Ethernet test is listed in your local test plan, it indicates that the test suite did not recognize the speed of that device. Check the connection before attempting to test that particular device.

Wireless test	Description
WirelessG	The network test with added speed detection for 802.11g wireless Ethernet connections.

Wireless test	Description
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.
WirelessAX (Superseded by WiFi6)	The network test with added speed detection for 802.11ax wireless Ethernet connections.
WiFi6	The network test with added speed detection for 802.11ac wireless Ethernet connections.
WiFi6E	The network test with added speed detection for 802.11ac wireless Ethernet connections.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The test runs the following subtests to gather information about all the network devices:

1. The bounce test on the interface is conducted using **nmcli conn up** and **nmcli conn down** commands (**ifup**,**ifdown** on RHEL 7).
2. If the root partition is not NFS or iSCSI mounted, the bounce test is performed on the interface. Additionally, all other interfaces that will not be tested are shut down to ensure that traffic is routed through the interface being tested.
3. If the root partition is NFS or iSCSI mounted, the bounce test on the interface responsible for the iSCSI or NFS connection is skipped, and all other interfaces, except for the one handling the iSCSI or NFS connection, will be shut down.
4. A test file gets created at location /**dev**/**urandom**, and its size is adjusted with the speed of your NIC.
5. TCP and UDP testing - The test uses iperf tool to:
    a. Test TCP latency between the test server and host under test. The test checks if the system runs into any OS timeouts and fails if it does.
    b. Test the bandwidth between the test server and the host under test. For wired devices, it is recommended that the speed is close to the theoretical maximum.
    c. Test UDP latency between the test server and host under test. The test checks if the system runs into any OS timeouts and fails if it does.

6. File transfer testing - The test uses SCP to transfer a file from the host under test to the remote system or test server and then transfers it back to the host under test to check if the transfer works properly.

7. ICMP (ping) test - The script causes a ping flood at the default packet size to ensure nothing in the system fails (the system should not restart or reset 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 retries 5 times for an acceptable success rate.

8. Finally, the test brings all interfaces back to their original state (active or inactive) when the test is executed.

Preparing for testing wired devices

You can test as many network devices as you want in each test run.

Before you begin:

- Ensure to connect each device at its native (maximum) speed, or else the test fails.
- Ensure that the test server is up and running.
- Ensure that each network device has an IP address assigned either statically or dynamically via DHCP.
- Ensure that multiple firewall ports are open, for the iperf tool to run TCP and UDP subtests.

NOTE

By default, ports 52001-52101 are open. If you want to change the default ports, update the **iperf-port** and **total-iperf-ports** values in the */etc/rhcert.xml* configuration file.

Example:

<server listener-port="8009" iperf-port="52001" total-iperf-ports="100">

If the firewall ports are not open, the test prompts to open the firewall ports during the test run.

Partitionable networking

The test checks if any of the network devices support partitioning, by checking the data transfer at full speed and the partitioning function.

Running the test based on the performance of NIC:

- If NIC runs at full speed while partitioned then, configure a partition with NIC running at its native speed and Perform the network test in that configuration.
- If NIC does not run at full speed while partitioned then, run the test twice - first time, run it without partitioning to see the full-speed operation, and the second time, run it with partitioning enabled to see the partitioning function.

NOTE

Red Hat recommends selecting either 1Gb/s or 10Gb/s for your partitioned configuration so that it conforms to the existing network speed tests.

Preparing for testing wireless Ethernet devices

Based on the wireless card that is being tested, the wireless access point that you connect to must have the capability to perform WirelessG, WirelessN, WirelessAC, WirelessAX, WiFi6, and WiFi6E network tests.

Executing the test

The network test is non-interactive. Run the following command and then select the appropriate **network** test name from the list that displays.

```
rhcert-run
```

Table A.3. Manually adding and running the test

Speed Type	Command to manually add Ethernet Test	Command to Manually run Ethernet Test
1GigEthernet	rhcert-cli plan --add --test 1GigEthernet --device <device name>	rhcert-cli run --test 1GigEthernet --server <test server IP addr>
10GigEthernet	rhcert-cli plan --add --test 10GigEthernet --device <device name>	rhcert-cli run --test 10GigEthernet --server <test server IP addr>
20GigEthernet	rhcert-cli plan --add --test 20GigEthernet --device <device name>	rhcert-cli run --test 20GigEthernet --server <test server IP addr>
25GigEthernet	rhcert-cli plan --add --test 25GigEthernet --device <device name>	rhcert-cli run --test 25GigEthernet --server <test server IP addr>
40GigEthernet	rhcert-cli plan --add --test 40GigEthernet --device <device name>	rhcert-cli run --test 40GigEthernet --server <test server IP addr>
50GigEthernet	rhcert-cli plan --add --test 50GigEthernet --device <device name>	rhcert-cli run --test 50GigEthernet --server <test server IP addr>

Speed Type	Command to manually add Ethernet Test	Command to Manually run Ethernet Test
100GigEthernet	rhcert-cli plan --add --test 100GigEthernet --device <device name>	rhcert-cli run --test 100GigEthernet --server <test server IP addr>
200GigEthernet	rhcert-cli plan --add --test 200GigEthernet --device <device name>	rhcert-cli run --test 200GigEthernet --server <test server IP addr>
400GigEthernet	rhcert-cli plan --add --test 400GigEthernet --device <device name>	rhcert-cli run --test 400GigEthernet --server <test server IP addr>

Replace **<device name>** and **<test server IP addr>** with the appropriate value.

Run time

The network test takes about 2 minutes to test each PCIe-based, gigabit, wired Ethernet card, and the required Supportable test adds about a minute to the overall run time.

Additional resources

- For more information about the remaining test functionality, see Ethernet test.

A.31. NETWORKMANAGEABLECHECK

What the test covers

The **NetworkManageableCheck** test runs for all the network interfaces available in the system.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test comprises two subtests that perform the following tasks:

1. Check the BIOS device name to confirm that the interface follows the terminology set by the firmware.

NOTE

BIOS device name validation runs only on x86 systems.

2. Check if the Network Manager manages the interface, for evaluating current network management status.

Executing the test

The **NetworkManageableCheck** test is mandatory. It is planned and executed with a self-check and **supportable** test to ensure thorough examination and validation of network interfaces.

Run time

The test takes around 1 minute to complete. However, the duration of the test varies depending on the specifics of the system and the number of interfaces.

A.32. NVME OVER FABRIC TESTS

NVMe over Fabrics, also known as NVMe-oF and non-volatile memory express over fabrics, is a protocol specification designed to connect hosts to storage across a network fabric using the NVMe protocol.

The protocol is designed to enable data transfers between a host computer and a target solid-state storage device or system over a network - accomplished through NVMe message-based commands. Data transfers can be transferred through methods such as Ethernet or InfiniBand.

A.32.1. nvme_infiniband

What the test covers

The **nvme_infiniband** test verifies the access and use of NVMe SSD drives over the RDMA network. The Host Under Test is configured as an NVMe client, and the lab agent system is configured as an NVMe target.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test runs multiple subtests to:

1. Verify that necessary kernel modules are loaded, and that the NVMe client is connected to the NVMe target.
2. Establish and confirm the connection between NVMe target and client by running discovery, disconnect, and connect commands.
3. Detect the network interface used to connect to the storage device in the target system and accordingly run one type of test from each of the STORAGE and infiniband connection tests.

For the NVMe over Fabric storage test, the test is executed on the NVMe client system but the NVMe device physically resides on the NVMe target host. Both NVMe client and target hosts communicate using the RDMA protocol.

Preparing for the test

Before you begin the test, ensure that:

- The NVMe target and NVMe client systems are configured properly and are part of the RDMA network.
- The NVMe client and NVMe target are running the same RHEL version. Otherwise, communication between the NVMe client on RHEL 9.0 and the NVMe target on RHEL 8.5 will result in an error similar to *Invalid MNAN value 1024 attempting nvme connect* .

Executing the test

The test is non-interactive. Currently, this test can be planned and executed via CLI only.

Run time

This test takes about 15 minutes to run. Any other mandatory or selected tests will add to the overall run time.

A.32.2. nvme_iwarp

What the test covers

The **nvme_iwarp** test verifies the access and use of NVMe SSD drives over the RDMA network. The test is supported to run on RHEL 8. The Host Under Test is configured as an NVMe client, and the lab agent system is configured as an NVMe target.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test runs multiple subtests to:

1. Verify that necessary kernel modules are loaded, and that the NVMe client is connected to the NVMe target.
2. Establish and confirm the connection between NVMe target and client by running discovery, disconnect, and connect commands.
3. Detect the network interface used to connect to the storage device in the target system and accordingly run one type of test from each of the STORAGE and iwarp connection tests.

For the NVMe over Fabric storage test, the test is executed on the NVMe client system but the NVMe device physically resides on the NVMe target host. Both NVMe client and target hosts communicate using the RDMA protocol.

Preparing for the test

Before you begin the test, ensure that the:

- NVMe client is running RHEL 8.x or 9.x.

- NVMe target and NVMe client systems are configured properly and are part of the RDMA network.
- NVMe client and NVMe target are running the same RHEL version, otherwise, communication between NVMe client on RHEL 9.0 to NVMe target on RHEL 8.5 will result in an error, *Invalid MNAN value 1024 attempting nvme connect*.

Executing the test

The test is non-interactive. Currently, this test can be planned and executed via CLI only.

Run time

This test takes about 15 minutes to run. Any other mandatory or selected tests will add to the overall run time.

A.32.3. nvme_omnipath

What the test covers

The **nvme_omnipath** test verifies the access and use of NVMe SSD drives over the RDMA network. The test is supported to run on RHEL 8. The Host Under Test is configured as an NVMe client, and the lab agent system is configured as an NVMe target.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test runs multiple subtests to:

1. Verify that necessary kernel modules are loaded, and that the NVMe client is connected to the NVMe target.
2. Establish and confirm the connection between NVMe target and client by running discovery, disconnect, and connect commands.
3. Detect the network interface used to connect to the storage device in the target system and accordingly run one type of test from each of the STORAGE and omnipath connection tests.

For the NVMe over Fabric storage test, the test is executed on the NVMe client system but the NVMe device physically resides on the NVMe target host. Both NVMe client and target hosts communicate using the RDMA protocol.

Preparing for the test

Before you begin the test, ensure that the:

- NVMe client is running RHEL 8.x or 9.x.
- NVMe target and NVMe client systems are configured properly and are part of the RDMA network.

- NVMe client and NVMe target are running the same RHEL version, otherwise, communication between NVMe client on RHEL 9.0 to NVMe target on RHEL 8.5 will result in an error, *Invalid MNAN value 1024 attempting nvme connect*.

Executing the test

The test is non-interactive. Currently, this test can be planned and executed via CLI only.

Run time

This test takes about 15 minutes to run. Any other mandatory or selected tests will add to the overall run time.

A.32.4. nvme_roce

What the test covers

The **nvme_roce** test verifies the access and use of NVMe SSD drives over the RDMA network. The test is supported to run on RHEL 8. The Host Under Test is configured as an NVMe client, and the lab agent system is configured as an NVMe target.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test runs multiple subtests to:

1. Verify that necessary kernel modules are loaded, and that the NVMe client is connected to the NVMe target.
2. Establish and confirm the connection between NVMe target and client by running discovery, disconnect, and connect commands.
3. Detect the network interface used to connect to the storage device in the target system and accordingly run one type of test from each of the STORAGE and RoCE connection tests.

For the NVMe over Fabric storage test, the test is executed on the NVMe client system but the NVMe device physically resides on the NVMe target host. Both NVMe client and target hosts communicate using the RDMA protocol.

Preparing for the test

Before you begin the test, ensure that the:

- NVMe client is running RHEL 8.x or 9.x.
- NVMe target and NVMe client systems are configured properly and are part of the RDMA network.
- NVMe client and NVMe target are running the same RHEL version, otherwise, communication between NVMe client on RHEL 9.0 to NVMe target on RHEL 8.5 will result in an error, *Invalid MNAN value 1024 attempting nvme connect*.

Executing the test

The test is non-interactive. Currently, this test can be planned and executed via CLI only.

Run time

This test takes about 15 minutes to run. Any other mandatory or selected tests will add to the overall run time.

A.32.5. nvme_tcp

What the test covers

The **nvme_tcp** test verifies the access and use of NVMe SSD drives over the TCP network. The test is currently available as a Technology Preview and is supported to run on RHEL 8. The Host Under Test is configured as an NVMe client, and the lab agent system is configured as an NVMe target.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test runs multiple subtests to:

1. Verify that necessary kernel modules are loaded, and that the NVMe client is connected to the NVMe target.
2. Establish and confirm the connection between NVMe target and client by running discovery, disconnect, and connect commands.
3. Detect the network interface used to connect to the storage device in the target system and accordingly run one type of test from each of the STORAGE and NETWORK tests.

For the NVMe over Fabric storage test, the test is executed on the NVMe client system but the NVMe device physically resides on the NVMe target host. Both NVMe client and target hosts communicate using the TCP protocol.

Preparing for the test

Before you begin the test, ensure that the:

- NVMe client is running RHEL 8.x.
- NVMe target and NVMe client systems are configured properly and are part of the RDMA network.

NOTE

The default TCP port number for NVMe over TCP is 8009. The default TCP port number for NVMe over RDMA is 4420. You can use any TCP port number that does not conflict with other current applications. If there is a port conflict, then reconfigure the NVMe port number 8009 with a different TCP port number.

Executing the test

The test is non-interactive. Currently, this test can be planned and executed via CLI only.

Run time

This test takes about 10 minutes to run. Any other mandatory or selected tests will add to the overall run time.

A.33. OMNIPATH CONNECTION

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 HUT to the selected IP address of the test server.
2. Rping test - Runs rping on test server and HUT using the selected test server IP address, then compares results to verify it ran to completion.
3. Rcopy test - Runs rcopy on test server and HUT, sending a randomly generated file and comparing md5sums on test server and HUT 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 spmquery on test server using device and port for another verification the device/port has been registered with the fabric.
9. ib_write_bw test
   Run ib_write_bw from the HUT to the selected IP address of the test server to test the Omnipath write bandwidth and verify if it can reach the required bandwidth. The queue pair parameter has been adjusted during the bandwidth test to achieve a throughput closer to the line rate.
10. ib_read_bw test
    Run ib_read_bw from the HUT to the selected IP address of the test server to test the Omnipath read bandwidth and verify if it can reach the required bandwidth. The queue pair parameter has been adjusted during the bandwidth test to achieve a throughput closer to the line rate.
11. ib_send_bw test
    Run ib_send_bw from the HUT to the selected IP address of the test server to test the Omnipath send bandwidth and verify if it can reach the required bandwidth. The queue pair

parameter has been adjusted during the bandwidth test to achieve a throughput closer to the line rate.

Preparing for the test

- Ensure that the test server and HUT are separate machines, on the same fabric. You need to install **opa-basic-tools** on the test server from the Downloads section of Red Hat customer portal web page.

NOTE

You must manually start the **httpd** service if you are using a RHEL 8 test server with a RHEL 7 host under test for running this test as the test suite does not start this service automatically.

Executing the test

This is an interactive test. Run the following command and then select the appropriate **omnipath connection** test name from the list that displays.

```
rhcert-run
```

You will be prompted with a dropdown to select an IP address (an IP address of test server) in which to perform the tests. Select an IP address corresponding to a device on the same fabric of the HUT device you are running the test for.

Manually adding and running the test

Use the following command to add the OmnipathConnectionTest manually:

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

Use the following command to manually run the OmnipathConnectionTest:

```
rhcert-cli run --test Omnipath --server <test server IP addr>
```

Run time

This test takes less than 10 minutes to run.

Additional resources

- For more information about InfiniBand and RDMA, see Understanding InfiniBand and RDMA technologies .

A.34. POWER_STOP

What the test covers

The Suspend-to-Idle state which, when enabled, allows a processor to be in the deepest idle state while the system is suspended. It freezes user space and puts all I/O devices into low-power states, thereby saving power consumption on systems.

The **power_stop** test is designed to verify if enabling these Stop (or idle) states work as expected on a ppc64le CPU architecture machine, specifically on Power9 based systems.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test uses the *lsprop* command to collect information of all the idle-stop states that a particular system supports, and *cpupower* command to enable and disable those states. The test observes the usage and duration counter increment of each cpu idle state to affirm if it's enabled or not.

Success Criteria:

Change in the usage and duration parameter values for the stop state before and after enabling it.

- PASS: If every state increases its counter parameter values
- WARN: If any one state fails to increase its counter parameter values
- FAIL: If any of the state does not increases its counter
- REVIEW: any other unknown issue

Preparing for the test

This test is planned automatically if the Host Under Test (HUT) meets the following requirements:

- The HUT is running one of the supported RHEL versions.
- The underlying architecture is ppc64le
- The CPU Model is POWER9

NOTE

This test is not supported and will fail when executed on any other RHEL version and architecture.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **power_stop** test name from the list that displays.

```
rhcert-run
```

Run time

The test takes less than five minutes to finish, but can vary depending on the number of CPU Idle Stop states.

A.35. PROFILER

The profiler test collects the performance metric from the Host Under Test and determines whether the metrics are collected from the software or the hardware Performance Monitoring Unit (PMU) supported by the RHEL Kernel. If the metrics are hardware-based, the test further determines if the PMU includes per core counters only or includes per package counters also. The profiler test is divided into three tests, **profiler_hardware_core**, **profiler_hardware_uncore**, and **profiler_software**.

A.35.1. profiler_hardware_core

What the test covers

The **profiler_hardware_core** test collects performance metrics using hardware-based per core counters by checking the cycle events. The core events measure the functions of a processor core, for example, the L2 cache.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The test is planned if core hardware event counters are found and locate the **cpu*cycles** files in the **/sys/devices** directory by running the **find /sys/devices/* -type f -name 'cpu*cycles'** command.

The test executes multiple commands to accumulate the sample of 'cycle' events, checks if the 'cpu cycle' event was detected, and checks if the samples were collected.

NOTE

This test is not intended to be exhaustive and, it does not test every possible core counter-event that a given processor may or may not have.

Preparing for the test

There are no special requirements to run this test.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **profiler_hardware_core** test name from the list that displays.

```
rhcert-run
```

Run time

The test takes approximately 30 seconds. Any other mandatory or selected tests will add to the overall run time.

A.35.2. profiler_hardware_uncore

What the test covers

The **profiler_hardware_uncore** test collects performance metrics using hardware-based package-wide counters. The uncore events measure the functions of a processor that are outside the core but are inside the package, for example, a memory controller.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test is planned if uncore hardware event counters are found. The test passes if it finds any uncore events and collects statistics for any one event. The test fails if it finds uncore events but does not collect statistics as those events are not supported.

The test executes multiple commands to collect the list of uncore events and the uncore events statistics.

NOTE

This test is not intended to be exhaustive and, it does not test every possible uncore counter-event that a given processor may or may not have.

Preparing for the test

There are no special requirements to run this test.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **profiler_hardware_uncore** test name from the list that displays.

```
rhcert-run
```

Run time

The test takes approximately 30 seconds. Any other mandatory or selected tests will add to the overall run time.

A.35.3. profiler_software

What the test covers

The **profiler_software** test collects performance metrics using software-based counters by checking the cpu_clock events.

Software counters can be certified using this test. However, for customers with high-performance requirements, this test can be limiting.

RHEL version supported

- RHEL 7
- RHEL 8

- RHEL 9

What the test does

The test is planned if no core hardware event counters are found.

The test executes multiple commands to accumulate the sample of cpu-clock events, checks if the cpu-clock event was detected, and checks if the samples were collected.

Preparing for the test

There are no special requirements to run this test.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **profiler_software** test name from the list that displays.

```
rhcert-run
```

Run time

The test takes approximately 30 seconds. Any other mandatory or selected tests will add to the overall run time.

A.36. REALTIME

What the test covers

The **realtime** test covers 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.

Additionally, for RHEL 8 and RHEL 9, the test ensures that there are cores reserved for housekeeping instead of fully utilizing all of them.

NOTE

The test is only planned on systems running Red Hat kernel.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 that 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. latency = t2 - (t1 + period)
5. goto 1

NOTE

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 and reports each datapoint.

While the **cyclictest** runs, 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.

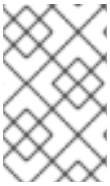
Additionally, for RHEL 8 and RHEL 9, the test checks if there are isolated CPUs configured in **/sys/devices/system/cpu/isolated** and the tuned version includes support for the initial auto setup of isolated_cores (version greater or equal to 2.19.0.). It also checks if the realtime tuned profile is active. If any check fails, the test gives a warning before continuing to execute.

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 realtime and will schedule the realtime test to run.
- For RHEL 8 and RHEL 9, with tuned version greater or equal to 2.19.0, select the tuned profile as realtime and reboot the system.

NOTE

If you need Realtime tuning assistance, then you must provide Red Hat access to your system to allow making required changes, including modifying BIOS.



NOTE

Newly installed kernels inherit the kernel command-line parameters from your previously configured kernels. For more information, see Changing kernel command-line parameters for all boot entries.

Executing the test

The realtime test is non-interactive. Run the following command and then select the appropriate **realtime** test name from the list that displays.

```
rhcert-run
```

The test will only appear when the system is running the rt-kernel.

Run time

The system management mode runs for two hours, and the timer event analysis runs for twelve hours on all machines. The required **supportable** test will add about a minute to the overall run time.

A.37. REBOOT

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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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. Run the following command and then select the appropriate **reboot** test name from the list that displays.

```
rhcert-run
```

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.38. ROCE CONNECTION

What the test does

The **RoCE 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 HUT to the selected IP address of the test server.
2. Rping test - Runs rping on test server and HUT using the selected test server IP address, then compares results to verify it ran to completion.

3. Rcopy test - Runs rcopy on test server and HUT, sending a randomly generated file and comparing md5sums on test server and HUT to verify successful transfer.

4. Ethtool test - Runs the ethtool command passing in the detected net device of the roce device.

5. ib_write_bw test
   Run ib_write_bw from the HUT to the selected IP address of the test server to test the RoCE write bandwidth and verify if it can reach the required bandwidth.

6. ib_read_bw test
   Run ib_read_bw from the HUT to the selected IP address of the test server to test the RoCE read bandwidth and verify if it can reach the required bandwidth.

7. ib_send_bw test
   Run ib_send_bw from the HUT to the selected IP address of the test server to test the RoCE send bandwidth and verify if it can reach the required bandwidth.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

Preparing for the test

- Ensure that the test server and HUT are separate machines, on the same fabric(s).

Executing the test

This is an interactive test. Run the following command and then select the appropriate **RoCE connection** test name from the list that displays.

```
rhcert-run
```

You will be prompted with a dropdown to select an IP address (an IP address of test server) in which to perform the tests. Select an IP address corresponding to a device on the same fabric of the HUT device you are running the test for.

Table A.4. Manually adding and running the test

Speed Type	Command to manually add RoCEConnection Test	Command to Manually run RoCEConnection Test
10GigRoCE	`rhcert-cli plan --add --test 10GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device>`	`rhcert-cli run --test 10GigRoCE --server <test server IP addr>`

Speed Type	Command to manually add RoCEConnection Test	Command to Manually run RoCEConnection Test
20GigRoCE	rhcert-cli plan --add --test 20GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 20GigRoCE --server <test server IP addr>
25GigRoCE	rhcert-cli plan --add --test 25GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 25GigRoCE --server <test server IP addr>
40GigRoCE	rhcert-cli plan --add --test 40GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 40GigRoCE--server <test server IP addr>
50GigRoCE	rhcert-cli plan --add --test 50GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 50GigRoCE --server <test server IP addr>
100GigRoCE	rhcert-cli plan --add --test 100GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 100GigRoCE --server <test server IP addr>

Speed Type	Command to manually add RoCEConnection Test	Command to Manually run RoCEConnection Test
200GigRoCE	rhcert-cli plan --add --test 200GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 200GigRoCE --server <test server IP addr>
400GigRoCE	rhcert-cli plan --add --test 400GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device>	rhcert-cli run --test 400GigRoCE --server <test server IP addr>

Replace **<device name>**, **<port number>**, **<net device>**, and **<test server IP addr>** with the appropriate values.

Additional resources

- For more information about InfiniBand and RDMA, see Understanding InfiniBand and RDMA technologies.

A.39. SATA

What the SATA test covers

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

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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.

Additional resources

- For more information about **what the test does** and **preparing for the test** see STORAGE.

A.40. 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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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

Additional resources

- For more information about **what the test does** and **preparing for the test** see STORAGE.

A.41. 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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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

Additional resources

- For more information about **what the test does** and **preparing for the test** see STORAGE.

A.42. 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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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

Additional resources

- For more information about **what the test does** and **preparing for the test** see STORAGE.

A.43. SAS

What the SAS test covers

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

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 mentions **SAS**

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

Additional resources

- For more information about **what the test does** and **preparing for the test** see STORAGE.

A.44. 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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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

Additional resources

- For more information about **what the test does** and **preparing for the test** see STORAGE.

A.45. 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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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

Additional resources

- For more information about **what the test does** and **preparing for the test** see STORAGE.

A.46. 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.

RHEL version supported

- RHEL 7
- RHEL 8

- RHEL 9

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

Additional resources

- For more information about **what the test does** and **preparing for the test** see STORAGE.

A.47. 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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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

Additional resources

- For more information about **what the test does** and **preparing for the test** see STORAGE.

A.48. NVDIMM

What the NVDIMM test covers

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

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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

Additional resources

- For more information about **what the test does** and **preparing for the test** see STORAGE.

A.49. SR-IOV

What the test does

The **SR-IOV** test certifies the NIC cards installed on the host under test (HUT) and the test server by checking if the SR-IOV functionality is supported on the cards.

The test is based on the Single Root I/O Virtualization (SR-IOV) technology that enables a single physical hardware device to be shared among multiple virtual machines or containers, improving the network performance and efficiency of I/O operations.

RHEL version supported

- RHEL 9

What the test covers

The test checks if the NIC card installed on the x86_64 system supports SR-IOV technology.

Preparing for the test

Before running the test:

- Install the NIC card on both the HUT and test server.
- Ensure that the NIC card undergoing testing has two ports each.
- Ensure that the test server and HUT are connected through two direct Ethernet cables back to back for the test to pass successfully.
- Ensure to enable Intel VT-d or AMD IOMMU and SR-IOV Global Enable parameters in your system's BIOS. For more details, refer to the system's BIOS configuration manual or any other methods provided by the manufacturer.
- Ensure to enable SRIOV for the NIC card under Device Settings. For more details, refer to the NIC card configuration manual or any other methods provided by the manufacturer.
- Install the following RPMs on both HUT and the test server, in the same order as mentioned.

  1. Enable the epel repository and then install beakerlib:

     ```
     # yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
     ```

     ```
     # yum install beakerlib
     ```

  2. SR-IOV

- Configure Hugepage

  ```
  # grubby --args='intel_iommu=on iommu=pt default_hugepagesz=1G hugepagesz=1G
  hugepages=32' --update-kernel=$(grubby --default-kernel)
  ```

  Ensure to reboot the system after configuration completes.

- Ensure to provision the HUT and test server. See Configuring the systems and running tests by using Cockpit or Configuring the systems and running tests by using CLI .

Executing the test

On HUT:

1. Edit the config file generated at this path according to your system's configuration, **/etc/redhat-certification/sriov/nic_cert.conf**

NOTE

You must update and keep a backup of the config file every time after running the provision command.

2. Run the test

```
# rhcert-run
```

While the test is executed on the HUT, a corresponding test run is also executed in auto mode on the test server. The test is non-interactive and runs in the background.

Run time

The test takes around 2 hours to run. Any other mandatory or selected tests will add to the overall run time.

A.50. 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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

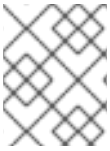
The STORAGE test performs the following actions on all storage devices with a unique parent:

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 **swapoff** 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 the **fio** or **dt** command is used to test the device. The **fio** or **dt** command is an I/O test program and is a generic test tool capable of testing, reading, and writing to devices. Multiple sets of test patterns verify the functionality of storage devices.

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.

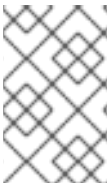


NOTE

The Storage test uses the **dt** package on RHEL 7.4 and older versions, and **fio** package on RHEL 7.5 and later versions.

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.



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. Run the following command and then select the appropriate **STORAGE** test name from the list that displays.

```
rhcert-run
```

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 **supportable** test will add about a minute to the overall run time.

Additional resources

- For more information about appropriate swap file sizing, see What is the recommended swap size for Red Hat platforms?.

A.51. SPECIAL KEYS

What the test covers

The **Special keys** test captures a variety of input events from the system integrated keyboard. This test runs on systems with batteries only.

RHEL version supported

- RHEL 8.6 and later
- RHEL 9

What the test does

The test captures the following:

- Non-ACPI-related signals such as volume up and down, volume mute, display backlight brightness up and down, and more.
- Key presses that send signals associated with global keyboard shortcuts, such as *<Meta+E>*, which opens the file browser.

Executing the test

The test is interactive. Run the following command and then select the appropriate **Special keys** test name from the list that displays.

```
rhcert-run
```

This test requires capturing all input events. During the test, press all the non-standard and multimedia keys on the device.

Press the Escape key at any time to end the test and see a list of keys. The test is successful if all the keys that you tested appear in the list.

Run time

The test takes less than 5 minutes to finish. Any other mandatory or selected tests will add to the overall run time.

A.52. SUPPORTABLE

What the test covers

The **supportable** test gathers basic information about the host under test (HUT). Red Hat uses this information to verify that the system complies with the certification requisites.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The test has several subtests that perform the following tasks:

1. Confirm that the **/proc/sys/kernel/tainted** file contains a zero (**0**), which indicates that the kernel is not tainted.

2. Confirm that package verification with the **rpm -V** command shows that no files have been modified.

3. Confirm that the **rpm -qa kernel** command shows that the buildhost of the kernel package is a Red Hat server.

4. Record the boot parameters from the **/proc/cmdline** file.

5. Confirm that the`rpm -V redhat-certification` command shows that no modifications have been made to any of the certification test suite files.

6. Confirm that all the modules shown by the **lsmod** command show up in a listing of the kernel files with the **rpm -ql kernel** command.

7. Confirm that all modules are on the Kernel Application Binary Interface (kABI) *stablelist*.

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 subtest tries to verify the kernel with data from the **redhat-certification** package. If the kernel is not present, the subtest attempts to verify the kernel by using the Internet connection.

   To verify the kernel by using the Internet connection, you must either configure the HUT's routing and DNS resolution to access the Internet or set the **ftp_proxy=http://proxy.domain:80** environment variable.

10. Check for any known hardware vulnerabilities reported by the kernel. The subtest reads the files in the **/sys/devices/system/cpu/vulnerabilities/** directory and exits with a warning if the files contain the word "Vulnerable".

11. Confirm if the system has any offline CPUs by checking the output of the **lscpu** command.

12. Confirm if Simultaneous Multithreading (SMT) is available, enabled, and active in the system.

13. Check if there is unmaintained hardware or drivers in systems running RHEL 8 or later.
    Unmaintained hardware and drivers are no longer tested or updated on a routine basis. Red Hat may fix serious issues, including security issues, but you cannot expect updates on any planned cadence.

    Replace or remove unmaintained hardware or drivers as soon as possible.

14. Check if there is deprecated hardware or drivers in systems running RHEL 8 or later.
    Deprecated hardware and drivers are still tested and maintained, but they are planned to become unmaintained and eventually disabled in a future release.

    Replace or remove deprecated devices or hardware as soon as possible.

15. Check if there is disabled hardware in systems running RHEL 8 or later.
    RHEL cannot use disabled hardware. Replace or remove the disabled hardware from your system before running the test again.

16. Run the following checks on the software RPM packages:

    - Check the RPM build host information to isolate non-Red Hat packages.

The test will ask you to explain the reasons for including the non-Red Hat packages. Red Hat will review the reasons and approve or reject each package individually.

- Check that the installed RPM packages are from the Red Hat products available in the offering and have not been modified.
  Red Hat reviews verification failures in the **rpm_verification_report.log** file. You will need to reinstall the failed packages and rerun the test.

17. Check the presence of both Red Hat and non-Red Hat firmware files in the system. It lists the non-Red Hat files, if present, and exits with REVIEW status.

18. Check the page size of systems by **getconf PAGESIZE** command.

After performing these tasks, the test gathers a *sosreport* and the output of the **dmidecode** command.

Executing the test

The **rhcert** tool runs the **supportable** test automatically as part of every run of the test suite. The **supportable** test runs before any other test.

The output of the **supportable** test is required as part of the test suite logs. Red Hat will reject test logs that do not contain the output of the **supportable** test.

Use the following command to run the test manually, if required:

```
$ rhcert-cli run -test supportable
```

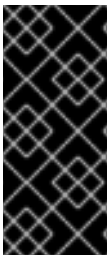
Run time

The **supportable** test takes around 1 minute on a 2013-era, single CPU, 3.3GHz, 6-core or 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 number of RPM files that are installed.

A.53. SUSPEND

What the test covers

(Laptops ony) The **suspend** test covers suspend/resume from S3 sleep state (suspend to RAM) and suspend/resume from S4 hibernation (suspend to disk). The test also covers the freeze (suspend to idle - s2idle) state that allows more energy to be saved. This test is only scheduled on systems that have built-in batteries, such as laptops.



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 **supportable** test, regardless of what tests are scheduled.

RHEL version supported

- RHEL 7

- RHEL 8
- RHEL 9

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.

Suspend states on RHEL 8 and RHEL 9 are written in the /**sys**/**power**/**state** file. RHEL 7 uses the **pm-utils** command instead.

- 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 kbd:[Fn]+kbd:[F-key] combination or dedicated kbd:[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 kbd:[Fn]+kbd:[F-key] combination or dedicated kbd:[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. Someone must be present at the Host Under Test in order to wake it up from suspend and hibernate.

Executing the test

The suspend test is interactive. Run the following command and then select the appropriate **suspend** test name from the list that displays.

```
rhcert-run
```

The test will prompt *suspend?* Answer **Yes** 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 host 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 Host 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 Host 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 Host 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 **supportable** test will add about a minute to the overall run time.

Additional resources

- For more information about appropriate swap file sizing, see What is the recommended swap size for Red Hat platforms?.

A.54. 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.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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. Run the following command and then select the appropriate **tape** test name from the list that displays.

```
rhcert-run
```

A.55. THUNDERBOLT3

What the test covers

The **Thunderbolt3** test covers Thunderbolt 3 ports from a hot plug and basic functionality standpoint, ensuring that all ports can be accessed by the OS and devices attached to the ports are properly added and removed.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The purpose of the test is to ensure that all Thunderbolt 3 ports present in a system function as expected. It asks for the number of available Thunderbolt3 ports and then asks the tester to plug and unplug a Thunderbolt 3 device into each port. The test watches for Thunderbolt 3 device 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. Note, while Thunderbolt 3 devices use the same physical connector as USB C devices, USB C devices are not Thunderbolt 3 devices. The test will not pass if USB C devices are used including USB C devices that claim compatibility with Thunderbolt 3 ports. Only Thunderbolt 3 devices can be used for this test.

Preparing for the test

Count the available Thunderbolt3 ports and have an available Thunderbolt3 device to use during the test.

Executing the test

The Thunderbolt3 test is interactive. Run the following command and then select the appropriate **Thunderbolt3** test name from the list that displays.

```
rhcert-run
```

When prompted by the system, enter the number of available Thunderbolt3 ports present on the system. The system will ask for a Thunderbolt3 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 Thunderbolt3 test takes about 15 seconds per Thunderbolt3 port. This includes the time to manually plug in the device, scan the port, unplug the device, and scan the port again. Any other mandatory or selected tests will add to the overall run time.

A.56. THUNDERBOLT4

What the test covers

The **Thunderbolt4** test covers Thunderbolt 4 ports from a hot plug and basic functionality standpoint, ensuring that all ports can be accessed by the OS and devices attached to the ports are properly added and removed.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The purpose of the test is to ensure that all Thunderbolt 4 ports present in a system function as expected. It asks for the number of available Thunderbolt4 ports and then asks the tester to plug and unplug a Thunderbolt 4 device into each port. The test watches for Thunderbolt 4 device 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. Note, while Thunderbolt 4 devices use the same physical connector as USB C devices, USB C devices are not Thunderbolt 4 devices. The test will not pass if USB C devices are used including USB C devices that claim compatibility with Thunderbolt 4 ports. Only Thunderbolt 4 devices can be used for this test. This test also validates that the generation of connection between the host and the connected device is Thunderbolt 4.

Preparing for the test

Count the available Thunderbolt4 ports and have an available Thunderbolt4 device to use during the test.

Executing the test

The Thunderbolt4 test is interactive. Run the following command and then select the appropriate **Thunderbolt4** test name from the list that displays.

```
rhcert-run
```

When prompted by the system, enter the number of available Thunderbolt4 ports present on the system. The system will ask for a Thunderbolt4 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 Thunderbolt4 test takes about 15 seconds per Thunderbolt4 port. This includes the time to manually plug in the device, scan the port, unplug the device, and scan the port again. Any other mandatory or selected tests will add to the overall run time.

A.57. USB_STORAGE

What the test covers

The **usb_storage** test adds speed detection functionality to the existing storage test. The usb_storage test comprises:

- **USB2_storage** test to detect the version of the connected USB device
- **USB3_storage** test to detect the version and interface speed of the connected USB device and supports multiple speeds (5Gbps, 10Gbps, 20Gbps, 40Gbps)

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test detects the interface speed and version of the USB device connected to the system and accordingly plans the corresponding test. For example, if a USB 3.0 device with a supported interface speed of 10Gbps is detected, the USB3_10Gbps_Storage subtest will be planned and executed.

Preparing for the test

Ensure that the USB storage device is connected to the system.

Executing the test

You can choose either way to run the test:

- Run the following command and then select the appropriate USB test name from the list that displays.

```
rhcert-run
```

- Run the **rhcert-cli** command by specifying the desired test name. For example,

```
rhcert-cli run --test=USB3_10Gbps_Storage
```

Additional resources

- For more information on the rest of the test functionality, see STORAGE.

A.58. 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.

RHEL version supported

- RHEL 7
- RHEL 8

- RHEL 9

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. Run the following command and then select the appropriate **USB2** test name from the list that displays.

```
rhcert-run
```

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 **supportable** test will add about a minute to the overall run time.

A.59. USB3

What the test covers

The **USB3** test covers USB3 ports from a basic functionality standpoint, ensuring that all ports can be enumerated, accessed, and hot plugged by the OS. The USB3 test supports three different speed-based tests, for each 5Gbps, 10Gbps, and 20Gbps. All three tests are planned if the system supports USB3. Successful credit for each test will result in the corresponding feature included in the Red Hat Ecosystem Catalog for the certification. The tests and their success criteria are as follows:

Success criteria:

- USB3_5Gbps - The test will pass when the device transfer speed is 5Gbps.
- USB3_10Gbps - The test will pass when the device transfer speed is 10Gbps.
- USB3_20Gbps - The test will pass when the device transfer speed is 20Gbps.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 an available USB3 device to use during the testing. You may need to trace the USB ports from the motherboard header(s) to distinguish between USB2 and USB3 ports. Ensure that the line speed of the device matches the expected speed of the test, that is, 5Gbps, 10 Gbps, or 20Gbps.

Executing the test

The USB3 test is interactive. Run the following command and then select the appropriate **USB3** test name from the list that displays.

```
rhcert-run
```

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 **supportable** test will add about a minute to the overall run time.

A.60. USB4

What the test covers

The **USB4** test covers USB4 ports from a basic functionality standpoint, ensuring that all ports can be enumerated, accessed, and hot plugged by the OS. The USB4 test supports two different speed-based tests, one for 20Gbps and one for 40Gbps. Both tests are planned if the system supports USB4. Successful credit for each test will result in the corresponding feature included in the Red Hat Ecosystem Catalog for the certification. The tests and their success criteria are as follows:

Success criteria

- USB4_20Gbps - The test will pass when the device transfer speed is 20Gbps.
- USB4_40Gbps - The test will pass when the device transfer speed is 40Gbps.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The purpose of the test is to ensure that all USB4 ports present in a system function as expected. It asks for the number of available USB4 ports (minus any that are in use for keyboard/mouse, etc.) and then asks the tester to plug and unplug a USB4 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 USB4 ports and have an available USB4 device to use during the testing. You may need to trace the USB ports from the motherboard header(s) to distinguish between USB2, USB3, and USB4 ports. Ensure that the line speed of the device matches the expected speed of the test, that is, 20Gbps or 40Gbps.

Executing the test

The USB4 test is interactive. Run the following command and then select the appropriate **USB4** test name from the list that displays.

```
rhcert-run
```

When prompted by the system, enter the number of available USB4 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 USB4 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 USB4 test takes about 15 seconds per USB4 port. This includes the time to manually plug in the device, scan the port, unplug the device, and scan the port again. The required **supportable** test will add about a minute to the overall run time.

A.61. VIDEO

What the test covers

For RHEL 7 and RHEL 8, the **VIDEO** test checks for all removable or integrated video hardware on the motherboard. Devices are selected for testing by their PCI class ID. Specifically, the test checks for a device with a PCI class as Display Controller in the *udev* command output.

For RHEL 9, the **VIDEO** test remains the same. However, for framebuffer graphic solutions, the test is planned after it identifies if the display kernel driver is in use as a framebuffer and if direct rendering is not supported using the **glxinfo** command.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The test runs multiple subtests:

1. Check Connections - Logs the **xrandr** command output. This subtest is optional, and its failure does not affect the overall test result.
2. Set Configuration - Checks the necessary configuration prerequisites like setting the display depth, flags, and configurations for the next subtest.
3. The X Server Test - Starts another display server using the new configuration file and runs the **glxgears**, a lightweight MESA OpenGL demonstration program to check the performance.
4. Log Module and Drivers - Runs **xdpyinfo** to determine the screen resolution and color depth. Along with that, the configuration file created at the start of the test should allow the system to run at the maximum resolution capability.

Finally, the test uses grep to search through the **/var/log/Xorg.0.log** logfile to determine in-use modules and drivers.

Preparing for the test

- Ensure that the monitor and video card in the system can run at a resolution of 1024x768 with a color depth of 24 bits per pixel (bpp). Higher resolutions or color depths are also acceptable. Check the **xrandr** command output for 1024x768 at 24 bpp or higher to confirm.
- If you do not see all the resolutions that the card or monitor combination can generate, ensure to remove any KVM switches between the monitor and video card.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **VIDEO** test name from the list that displays.

```
rhcert-run
```

First, the test system screen will go blank, and then a series of test patterns from the x11perf test program will appear. When the test finishes, it will return to the desktop or the virtual terminal screen.

Run time

The test takes about 1 minute to complete. Any other mandatory or selected tests will add to the overall run time.

A.62. VIDEO_PORTS

What the test covers

The **VIDEO_PORTS** test checks whether all the graphics outputs ports of each graphics processor in the system are functioning.

The test runs on machines that have one or more graphics output ports. Machines with one or more embedded or add-on graphics processors are also supported, including laptops with ports wired to integral panels.

The test does not run on a port if it does not detect a display connected to that port.

RHEL version supported

- RHEL 9

What the test does

The test performs the following actions:

1. The test runs through each port that it detects as having a monitor connected.
2. The test then launches a glmark2 window and prompts you to drag the test window to each connected display.
3. If the test detects additional ports that are untested, it goes into interactive mode. It prompts you to attach a display to each untested port and to repeat the test.
4. The test continues to run in this loop until it has tested all detected ports, or until you indicate that the untested ports are not usable by customers. If there are unusable ports, the test prompts you for clarification.
5. When the loop exits, the test displays a PASS result if all ports have been tested or a REVIEW result if some ports were identified as unusable.

Preparing for the test

- Prepare a set of monitors that have the appropriate connectors for your system. This includes a built-in monitor and at least one external monitor.
- If there are less monitors than ports, the test will run in loops and allow you to connect the displays to ports in batches. The built-in monitor must continue working in addition to each of the external monitors attached.
- There may be more electronic than physical ports, meaning that the hardware supports more displays than the system makes available to the user. The list of ports displayed on the screen when the test begins is not relevant to the test.
- There may be more physical ports in the system than can be used all at once. There may also be ghost ports such as service ports or USBs. You must be able to differentiate between a port that is not functioning due to incompatibility with another port or because it is a ghost port, and between a port that is not functioning at all.

Executing the test

The **VIDEO_PORTS** test is interactive. Before executing the test, connect a monitor to at least one of the graphics output ports.

1. Provision the system:
   a. Run this command:

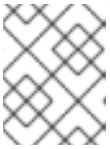
```
# rhcert-provision
```

b. When prompted, enter the path of the test plan saved on your system.

c. If prompted, provide the hostname or the IP address of the test server to set up a passwordless SSH. You will only be prompted the first time you add a new system.

2. Start the test:

```
# rhcert-cli run --test VIDEO_PORTS
```



NOTE

The test starts by listing a set of internal displays, both connected and disconnected. These do not represent the physical ports being tested.

3. For each connected graphics output port, follow the steps below:

a. Wait for the test to identify the port. When prompted, press any key to continue.

b. The **glmark2** window opens. Move this window to the monitor connected to the port, if different from the active monitor.
The glmark2 benchmark measures various aspects of OpenGL (ES) 2.0 performance on the identified display. The benchmark invokes a series of images, which test different combinations of surface, angle, color, and light.

c. Wait for the **glmark2** window to close. You will see a glmark2 score and a **Test passed** message for each successful test.

4. For each unconnected graphics output port, follow the steps below:

a. When prompted, connect a monitor to the graphics port and enter **yes** to continue.
On the first prompt, you can enter **no** to end the GRAPHICS_PORTS test. For each additional prompt, a timer is displayed that gives you 20 seconds to connect the monitor. The timer is repeated three times before timeout.

b. Wait for the test to identify the port. When prompted, press any key to continue.

c. Move the **glmark2** window to the monitor connected to the port. Wait for the window to close.

5. When there are no additional ports to connect, let the timer run for 60 seconds until timeout. The test exits with a PASS result if all ports were tested successfully.

6. Optionally save the test results to a log file:

```
# rhcert-save
```

7. Access the log file from your browser, by navigating to the location of the log files.

Run time

The test time varies according to the number of ports being tested. Each port takes around 2-3 minutes to test.

Additional factors impacting test time include system performance, such as memory frequency and CPU. Any other mandatory or selected tests will add to the overall run time.

A.63. VIDEO_DRM

What the test covers

The **VIDEO_DRM** test verifies the graphics controller, which utilizes a native DRM kernel driver with basic graphics support.

The test will plan if:

- The display driver in use is identified as a kernel mode-setting driver.
- The display driver is not a framebuffer.
- The direct rendering is not supported as identified by the **glxinfo** command, and the OpenGL renderer string is **llvmpipe**.

RHEL version supported

- RHEL 9

What the test does

The test verifies the functionality of the graphics controller similar to the VIDEO test.

Preparing for the test

- Ensure that the monitor and video card in the system can run at a resolution of 1024x768 with a color depth of 24 bits per pixel (bpp). Higher resolutions or color depths are also acceptable. Check the **xrandr** command output for 1024x768 at 24 bpp or higher to confirm.
- If you do not see all the resolutions that the card or monitor combination can generate, ensure to remove any KVM switches between the monitor and video card.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **VIDEO_DRM** test name from the list that displays.

```
rhcert-run
```

First, the test system screen will go blank, and then a series of test patterns from the x11perf test program will appear. When the test finishes, it will return to the desktop or the virtual terminal screen.

Run time

The test takes about 1 minute to complete. Any other mandatory or selected tests will add to the overall run time.

A.64. VIDEO_DRM_3D

What the test covers

The **VIDEO_DRM_3D** test verifies the graphics controller, which utilizes a native DRM kernel driver with accelerated graphics support.

The test will plan if:

- The display driver in use is identified as a kernel mode-setting driver.
- The display driver is not a framebuffer.
- The direct rendering is supported as identified by the **glxinfo** command, and the OpenGL renderer string is not **llvmpipe**.

The test uses Prime GPU Offloading technology to execute all the video test subtests.

RHEL version supported

- RHEL 9

What the test does

The test verifies the functionality of the graphics controller similar to the VIDEO test. In addition, the test runs the following subtests:

1. Vulkaninfo test - Logs the **vulkaninfo** command output to collect the Vulkan information such as device properties of identified GPUs, Vulkan extensions supported by each GPU, recognized layers, supported image formats, and format properties.
2. Glmark2 benchmarking test - Runs the **glmark2** command to generate the score based on the OpenGL 2.0 & ES 2.0 benchmark set of tests and confirms the 3D capabilities. The subtest executes the utility two times with a different set of parameters, first with the Hardware renderer and later with the Software renderer. If the Hardware renderer command-run results in a better score than software, the test passes successfully, confirming the display controller has better 3D capabilities, otherwise fails.

Preparing for the test

- Ensure that the monitor and video card in the system can run at a resolution of 1024x768 with a color depth of 24 bits per pixel (bpp). Higher resolutions or color depths are also acceptable. Check the **xrandr** command output for 1024x768 at 24 bpp or higher to confirm.
- If you do not see all the resolutions that the card or monitor combination can generate, ensure to remove any KVM switches between the monitor and video card.

Executing the test

The test is non-interactive. Run the following command and then select the appropriate **VIDEO_DRM_3D** test name from the list that displays.

```
rhcert-run
```

First, the test system screen will go blank, and then a series of test patterns from the x11perf test program will appear. When the test finishes, it will return to the desktop or the virtual terminal screen.

Run time

The test takes about 1 minute to complete. Any other mandatory or selected tests will add to the overall run time.

A.65. WIRELESSG

What the test covers

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

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 minimum throughput of 22Mb/s in order to pass.

Additional resources

- For more information on the rest of the test functionality, see network.

A.66. WIRELESSN

What the test covers

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

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 minimum throughput of 100Mb/s in order to pass.

Additional resources

- For more information on the rest of the test functionality, see network.

A.67. WIRELESSAC

What the test covers

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

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

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 minimum throughput of 300Mb/s in order to pass.

Additional resources

- For more information about the rest of the test functionality, see network.

A.68. WIRELESSAX (SUPERSEDED BY WIFI6)

What the test covers

The **WirelessAX** test is run on all wireless Ethernet connections with a maximum connection speed of 802.11ax.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The test detects "ax" link type reported by **iw** and matches the product name having "wifi 6" or "AX" to decide the device has AX Support. The test for Wireless AX is also planned if the device passes Wireless AC Test and demonstrates a minimum throughput of 1200 Mb/s in order to pass. This test is not planned automatically but can be planned manually via CLI. Instead, WiFi6 test is planned automatically.

A.69. WIFI6

What the test covers

The **WiFi6** test is run on all wireless Ethernet connections with a maximum connection speed of 802.11ax.

RHEL version supported

- RHEL 7
- RHEL 8
- RHEL 9

What the test does

The test detects "ax" link type reported by **iw** and matches the product name having "wifi 6" or "AX" to decide if the device has AX Support. The test for WiFi6 is also planned if the device passes Wireless AC Test and demonstrates a minimum throughput of 1200 Mb/s in order to pass.

A.70. WIFI6E

What the test covers

The **WiFi6E** test is run on all wireless Ethernet connections with a maximum connection speed of 802.11ax utilizing the 6GHz frequency band.

RHEL version supported

- RHEL 8
- RHEL 9

What the test does

The test detects "ax" link type reported by **iw** and matches the product name containing "wifi 6E" or "AX" to decide if the device has AX Support. The test for WiFi6E is also planned if the device passes Wireless AC Test and demonstrates a minimum throughput of 6000 Mb/s in order to pass.

A.71. MANUALLY ADDING AND RUNNING THE 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 HUT. Any modifications you make to the local test plan will be sent to the test server, so you can continue to use the web interface on the test server 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_live_migration
- hwcert/fvtest/fv_memory
- hwcert/fvtest/fv_network
- hwcert/fvtest/fv_storage
- hwcert/fvtest/fv_pcie_storage_passthrough
- hwcert/fvtest/fv_pcie_network_passthrough
- hwcert/fvtest/fv_usb_storage_passthrough
- hwcert/fvtest/fv_usb_network_passthrough
- hwcert/fvtest/fv_cpu_pinning
- hwcert/profiler
- hwcert/storage
- hwcert/video
- hwcert/supportable
- hwcert/optical/bluray
- hwcert/optical/dvd
- hwcert/optical/cdrom
- hwcert/fencing

- hwcert/realtime
- hwcert/reboot
- hwcert/tape
- hwcert/rdma/Infiniband_QDR
- hwcert/rdma/Infiniband_FDR
- hwcert/rdma/Infiniband_EDR
- hwcert/rdma/Infiniband_HDR
- hwcert/rdma/Infiniband_Socket_Direct
- hwcert/rdma/10GigRoCE
- hwcert/rdma/20GigRoCE
- hwcert/rdma/25GigRoCE
- hwcert/rdma/40GigRoCE
- hwcert/rdma/50GigRoCE
- hwcert/rdma/100GigRoCE
- hwcert/rdma/200GigRoCE
- hwcert/rdma/10GigiWarp
- hwcert/rdma/20GigiWarp
- hwcert/rdma/25GigiWarp
- hwcert/rdma/40GigiWarp
- hwcert/rdma/50GigiWarp
- hwcert/rdma/100GigiWarp
- hwcert/rdma/200GigiWarp
- hwcert/rdma/Omnipath
- hwcert/network/Ethernet/2_5GigEthernet
- hwcert/network/Ethernet/5GigEthernet
- hwcert/network/Ethernet/20GigEthernet
- hwcert/network/Ethernet/25GigEthernet-
- hwcert/network/Ethernet/50GigEthernet
- hwcert/network/Ethernet/100GigEthernet

    - hwcert/network/Ethernet/200GigEthernet
    - rhcert/self-check
    - hwcert/sosreport
    - hwcert/storage/U2 SATA
    - hwcert/storage/M2 SATA
    - hwcert/storage/SATA_SSD
    - hwcert/storage/SATA
    - hwcert/storage/SAS_SSD
    - hwcert/storage/SAS
    - hwcert/storage/U2_NVME
    - hwcert/storage/M2_NVME
    - hwcert/storage/PCIE_NVME
    - hwcert/storage/NVDIMM
    - hwcert/storage/STORAGE
- 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.
- Run the rhcert-cli command by specifying the test name:

```
rhcert-cli run --test=<test_name>
```

  for example:

```
rhcert-cli run --test=audio
```

- You can specify **--device** to run the specific device:

```
rhcert-cli run --test=<test name> --device=<device name>
```

  for example:

```
rhcert-cli run --test=kdump --device=nfs
```

NOTE

It is advisable to use **rhcert-cli** or **rhcert-run** independently and save the results. Mixing the use of both **rhcert-cli** and **rhcert-run** and saving the results together may result in the inability to process the results correctly.

Revised on 2024-06-24 21:06:24 UTC