

Red Hat Enterprise Linux Hardware Certification Test Suite User Guide

The Guide to Performing Red Hat Hardware Certification



Gary Case

Red Hat Enterprise Linux Hardware Certification Test Suite User Guide

The Guide to Performing Red Hat Hardware Certification Edition 1.1

Author

Gary Case

gcase@redhat.com

Copyright © 2013 | Red Hat, Inc. |.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](http://creativecommons.org/licenses/by-sa/3.0/)¹. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

The Red Hat Enterprise Linux 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, then explains how to set up the certification environment, test the systems or components being certified, and submit the results to Red Hat for verification. Background information is also provided, including the test methodology and results evaluation.

¹ <http://creativecommons.org/licenses/by-sa/3.0/>

Preface	v
1. Document Conventions	v
1.1. Typographic Conventions	v
1.2. Pull-quote Conventions	vi
1.3. Notes and Warnings	vii
2. Getting Help and Giving Feedback	vii
2.1. Do You Need Help?	vii
2.2. We Need Feedback	viii
 1. The Red Hat Hardware Certification Program	 1
1.1. Introduction	1
1.2. Getting Help	1
1.3. Certification Process Summary	1
1.4. Establishing a Relationship with Red Hat	2
 2. Preparing to Test	 5
2.1. Choosing the Type of Certification	5
2.2. Opening a Standard Certification Request	5
2.3. NEW - Opening a Hardware Specification Entry	14
2.4. NEW - Opening Standard Certification Requests from a Hardware Specification Entry	15
2.5. Opening a Supplemental Certification Request	17
2.6. NEW - Opening a Supplemental Certification Request from a Hardware Specification Entry	18
2.7. Opening a System Pass-Through Certification Request	19
2.8. NEW - Opening a System Pass-Through Certification Request from a Hardware Specification Entry	20
2.9. Submitting Specs When no Public URL Exists	23
2.10. How Red Hat Creates the Official Test Plan	25
2.11. Leveraging of Existing Test Results	27
2.12. Preparing the Test Environment	28
2.13. Preparing a Kickstart File to Install RHEL 5 and hwcert-client	35
2.14. Preparing a Kickstart File to Install RHEL 6 and hwcert-client	37
2.15. Preparing a Kickstart File to Install RHEL 6 and hwcert-server	38
2.16. UPDATED - Using Locally-hosted Pre-built Guest Files	39
2.17. Configuring the network and kdump Test Server	40
2.18. Configuring the System Under Test	41
 3. Testing Systems and Components	 45
3.1. Scheduling and Running Tests with hwcert	45
3.1.1. A Few Words on screen	45
3.1.2. Automated Testing of a New System	45
3.1.3. Continuing Testing of a Partially Complete System	49
3.1.4. Editing the Local Test Plan	50
3.2. Certifying for Red Hat Enterprise Linux	51
3.2.1. Testing a Server	51
3.2.2. Testing a Desktop/Workstation	53
3.2.3. Testing a Laptop	54
3.2.4. Testing a Component	56
3.3. Certifying for Red Hat Enterprise Linux Open Stack Platform Compute	57
3.4. Certifying for Red Hat Enterprise MRG Realtime	57
3.5. Certifying for Red Hat Storage	59
3.6. Troubleshooting Failed Tests	59
3.7. Submitting Results	60
3.7.1. Automatic Upload from the SUT	60
3.7.2. Manual Upload	61

3.7.3. Automatic Upload from the Local Test Server	64
3.8. Monitoring Certification Progress	64
3.9. Completing Certifications	65
A. Appendixes	67
A.1. Test Procedures	67
A.1.1. NEW - 1GigEthernet (RHEL 6 only)	67
A.1.2. NEW - 10GigEthernet (RHEL 6 Only)	67
A.1.3. NEW - 100MegEthernet (RHEL 6 Only)	67
A.1.4. audio	68
A.1.5. battery	69
A.1.6. bluray	70
A.1.7. cdrom	73
A.1.8. core	73
A.1.9. cpuscaling	74
A.1.10. dvd	76
A.1.11. NEW - Ethernet (RHEL 6 Only)	76
A.1.12. expresscard	78
A.1.13. fencing (Optional)	78
A.1.14. fv_core	79
A.1.15. fv_memory	79
A.1.16. fv_network (Optional)	79
A.1.17. fv_storage (Optional)	79
A.1.18. hwlatdetect	79
A.1.19. info	80
A.1.20. kdump	80
A.1.21. lid	81
A.1.22. memory	81
A.1.23. network	83
A.1.24. pccard	86
A.1.25. profiler	86
A.1.26. reboot (Optional)	87
A.1.27. rteval	87
A.1.28. storage	88
A.1.29. suspend	89
A.1.30. tape	90
A.1.31. usb (RHEL 5 Only)	90
A.1.32. NEW - USB2 (RHEL 6 only)	91
A.1.33. NEW - USB3 (RHEL 6 only)	91
A.1.34. video	92
A.1.35. NEW - WirelessG (RHEL 6 only)	92
A.1.36. NEW - WirelessN (RHEL 6 only)	93
A.1.37. wlan (RHEL 5 Only)	93
B. Revision History	95
Index	97

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the [Liberation Fonts](https://fedorahosted.org/liberation-fonts/)¹ set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later include the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

Press **Enter** to execute the command.

Press **Ctrl+Alt+F2** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **mono-spaced bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select the **Left-handed mouse** check box and click

¹ <https://fedorahosted.org/liberation-fonts/>

Close to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — *username*, *domain.name*, *file-system*, *package*, *version* and *release*. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                                       struct kvm_assigned_pci_dev *assigned_dev)
{
    int r = 0;
    struct kvm_assigned_dev_kernel *match;
```

```

mutex_lock(&kvm->lock);

match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                             assigned_dev->assigned_dev_id);
if (!match) {
    printk(KERN_INFO "%s: device hasn't been assigned before, "
               "so cannot be deassigned\n", __func__);
    r = -EINVAL;
    goto out;
}

kvm_deassign_device(kvm, match);

kvm_free_assigned_device(kvm, match);

out:
mutex_unlock(&kvm->lock);
return r;
}

```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.



Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.



Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. Getting Help and Giving Feedback

2.1. Do You Need Help?

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal at <http://access.redhat.com>. Through the customer portal, you can:

- search or browse through a knowledgebase of technical support articles about Red Hat products.
- submit a support case to Red Hat Global Support Services (GSS).
- access other product documentation.

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at <https://www.redhat.com/mailman/listinfo>. Click on the name of any mailing list to subscribe to that list or to access the list archives.

2.2. We Need Feedback

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you. Please submit a report in Bugzilla: <http://bugzilla.redhat.com/> against the product **Documentation**.

When submitting a bug report, be sure to mention the manual's identifier:

Hardware_Certification_User_Guide

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

The Red Hat Hardware Certification Program

1.1. Introduction

The Red Hat Hardware Certification Program is Red Hat's method of certifying hardware to be compatible with Red Hat Enterprise Linux (RHEL) and other Red Hat software products. The program has three aspects: the hardware test suite, called **hwcert-client**, the Red Hat Hardware Catalog, located at <http://hardware.redhat.com>, and the establishment of a joint support relationship between Red Hat and the vendor. The hwcert suite contains a series of basic functionality tests that exercise all areas of the system (RAM, storage, video, etc.). If all the tests pass, the hardware can be listed in the Red Hat Hardware Catalog, which is Red Hat's source for listing all hardware that has been tested and proven to work with Red Hat software products. The joint support relationship that we establish during the initial certification process assures our mutual customers that if a problem is discovered down the line, we have the necessary agreements and contacts in place to jointly work the problem until a resolution is found.

This user guide covers all aspects of the Red Hat Hardware Certification process, from the initial request for certification to the final approval and posting of the cert. It is geared towards the engineer who has been tasked with certifying his or her company's systems to run Red Hat Enterprise Linux or other Red Hat software product. It assumes that you as the tester have a basic level of knowledge about the software that is being used for certification, and are familiar with concepts like operating system installation and hardware installation and removal.

The policies and other rules of the program are covered in the separate Red Hat Hardware Certification Policy Guide, which is available at the following link:

https://access.redhat.com/site/documentation/en-US/Red_Hat_Hardware_Certification/1/html-single/Policy_Guide/index.html

1.2. Getting Help

Help is available should you run into trouble during any part of the certification process. If you have general questions about the certification procedure or encounter problems regarding the Red Hat Hardware Certification program, its requirements, policies, test suite, catalog, etc., you should open a ticket with Red Hat's Global Support Services. Refer to https://access.redhat.com/support/policy/support_process.html for information about this process. Partners who have been assigned a TAM can use Red Hat's Bugzilla site (<http://bugzilla.redhat.com>) to open a ticket. If questions related to the certification of a system or component arise during certification procedure, add a comment to the certification request to get help from the review team. You can also check [Section 3.6, "Troubleshooting Failed Tests"](#) for advice on common test failures. For all other questions (Sales, Customer Service, etc.), please see the Red Hat Contact page at <http://www.redhat.com/about/contact/> to determine how to contact the appropriate party.

1.3. Certification Process Summary

We begin with an outline of the entire certification procedure. We will discuss each step in greater detail as we move through the document.

1. The partner establishes a certification relationship with Red Hat.

2. The partner creates a request for the certification of a specific model of computer or hardware component on Red Hat's hardware certification website (<http://hardware.redhat.com>¹).
3. Red Hat's certification review team creates an official test plan inside the certification request to track the testing of all relevant components.
4. The partner runs the tests specified in the official test plan and submits results packages to Red Hat for analysis.
5. The review team analyzes the test results and marks them as completed when they receive passing results. Any failures require retesting.
6. The partner provides Red Hat with a representative hardware sample that covers the items that are being certified.
7. The system or component is marked as certified when all tests have passing results, and the entry is made visible to the public on the Red Hat Hardware Certification website if the partner requested a published certification.

You can refer to [Section 1.2, "Getting Help"](#) if you need help at any time.

1.4. Establishing a Relationship with Red Hat

Red Hat requires a relationship agreement that is specific to hardware certification before we can accept certification requests. This relationship may be documented in a section or subsection of an OEM or other partnership agreement, or it can be established by an independent hardware certification agreement. The creation of an OEM or other partnership agreement is not covered here, as that is something handled by a member of Red Hat's sales team. Please contact a member of the sales team if you are interested in incorporating certification into your existing agreement(s). If you would like to establish an independent hardware certification agreement, you can do so via the following steps:

1. Create an account with the Red Hat Hardware Program
 - a. Navigate to <http://www.redhat.com/partners/hardwarepartners/>
 - b. Click the **Buy Now** button and follow the steps. The account created during this step is also used on the [Red Hat Customer Portal](#)² and [Red Hat Network \(RHN\)](#)³.
2. Create a Red Hat Bugzilla account
 - a. Navigate to <https://bugzilla.redhat.com/>
 - b. Select **New Account** in the menu and follow the instructions. The account created during this step is also used to sign in to the [Red Hat Hardware Catalog](#)⁴ when working with certification requests.
3. Obtain required Red Hat Hardware Catalog permissions
 - File a support request to have certification creation permissions added to your account. If you have an assigned Technical Account Manager (TAM), file a request through Red Hat

¹ <http://hardware.redhat.com/>

² <http://access.redhat.com>

³ <http://rhn.redhat.com>

⁴ <http://hardware.redhat.com>

Bugzilla at <http://bugzilla.redhat.com>. If you do not have a TAM, file a request through Red Hat's Customer Portal at <http://access.redhat.com>. Include your Red Hat Hardware Program and Bugzilla user names (but not the passwords!) in the request. After this request has been approved and you log in to the hardware catalog, the **Create** link will appear in the hardware catalog menu. This allows you to create a new certification request.

Preparing to Test

2.1. Choosing the Type of Certification

There are several different types of certifications that exist within the Red Hat Hardware Certification Program:

- **Published or Unpublished Standard Certification** - This type of certification is the most common one, where a previously untested system or component is tested and proven to be compatible with Red Hat Enterprise software products. In the published case, the device is published on the hardware catalog <http://hardware.redhat.com> after certification is complete. In the unpublished case, the device is kept private and does not appear publicly on the hardware catalog.
- **Supplemental Certifications** - This type of certification covers the testing of changes that have been made in already certified equipment, such as a processor family upgrade in an existing certified server. It remains unpublished when completed.
- **System Pass-Through Certification** - This type of certification is performed when a system is being resold under a different vendor, make, or model name. It is also used when a vendor sells a series of systems whose configurations are subsets of a larger, superset system.

We will go through the procedure for opening a standard certification first, then explain the differences between standard and other certification types in the sections that follow.

2.2. Opening a Standard Certification Request

In a standard certification, systems or components are tested following the steps in [Section 1.3, "Certification Process Summary"](#), resulting in a published certification that is viewable by any member of the general public who visits the hardware catalog. The option also exists for an unpublished certification, which is typically used in the case of models or components that are not available to the general public. All the steps are identical in published and unpublished certifications until the last step is reached, when the certification reviewer chooses to either publish a certification and make it visible to the public, or leave it unpublished and make it visible to only the original partner and Red Hat.



Important

Published and unpublished certifications are covered by the same [policy guide](#)¹ rules. Unpublished certifications cannot be used to work around problems that render a system uncertifiable, such as non-functional integrated hardware or storage controllers that all require proprietary drivers for functionality. See the policy guide for more information.

Examples of completed published and unpublished certifications for the hypothetical "Mycompany Ultraserver X12" and "Mycompany Ultraserver X16" servers can be seen in the figures below. Notice how the unpublished certifications, which are indicated by a "lock" icon, are only visible in the partner view, which is what a properly-entitled Red Hat Hardware Certification Partner sees after logging in to the hardware catalog.

¹ https://access.redhat.com/site/documentation/en-US/Red_Hat_Hardware_Certification/1/html-single/Policy_Guide/index.html

Systems

Products	Version 6		Version 5		Version 4	
	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit
Mycompany Myserver X12						
Red Hat Enterprise Linux						4.8 ¹
Mycompany Myserver X16						
Red Hat Enterprise Linux		6.1		5.7 ¹		
Red Hat OpenStack Compute		3.0				

Figure 2.1. Public View of Published Certifications

Systems



Products	Version 6		Version 5		Version 4	
	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit
Mycompany Myserver X12						
Red Hat Enterprise Linux		 6		 5.3 ¹		4.8 ¹
Mycompany Myserver X16						
Red Hat Enterprise Linux		6.1		5.7 ¹		
Red Hat OpenStack Compute		3.0				

Figure 2.2. Partner View of Published and Unpublished Certifications

The certification processes for all the different types of certifications are managed through the Red Hat Hardware Catalog, located at <http://hardware.redhat.com>. We will go through the standard certification process in this example.



Note

The **hwcert** command can open a new certification request from the command line of the system under test. We recommend that users who are new to the program follow the steps in this section at least once to become familiar with the whole process before they use the automatic creation function built in to **hwcert**. If you are already familiar with the process, please see [Section 3.1.2, “Automated Testing of a New System”](#) for information on opening new certification requests with **hwcert**.

To begin, use the Red Hat Bugzilla account that you created in [Section 1.4, “Establishing a Relationship with Red Hat”](#) to log in to the catalog website by clicking the **Vendor Login** link on the

site. Click the **Create** link in the menu bar after logging in to start the creation process. The page that loads contains a number of fields for you to fill out:

Create New Entry

Reporter

tester@mycompany.com

Entry Type

Create Both

Product Data Source

☐ Acquire from Package

☒ Provide Manually

Create New Entry Fields

- **Reporter** - This field is automatically filled in with your username.
- **Entry Type** - This field lets you choose what kind of entry to create. You can create a hardware certification request, a hardware specification entry, or both types of requests. We recommend choosing *Create Both* to create a hardware certification request and the hardware specification entry. The use of Hardware Specification entries is optional at this time. Choose *Hardware Certification* if you only want to create a standard certification request. For more information on hardware specification entries, please see [Section 2.3, "NEW - Opening a Hardware Specification Entry"](#).
- **Product Data Source** - The catalog can read data from a results package to fill out some of the fields on this page. Selecting *Acquire from Package* and providing an appropriate results package will automatically populate the Make, Model, Version and Platform fields in the next sections. This option may be utilized after you are familiar enough with the certification process to feel comfortable selecting hardware and creating results packages prior to opening the certification request. Until that time, we recommend using the *Provide Manually* option and filling out those fields as described below, delaying the upload of any results packages until after the official test plan has been created.

UPLOAD CERTIFICATION PACKAGE

File

SELECT FILE

Currently supported test suite versions: **hwcert-dient-1.5.10-2**, **hwcert-dient-1.6.4-57**, **hwcert-dient-1.5.9-37**, v

Currently acceptable maximum test suite result file size: **16MB**

Description

Describe the attachment briefly.

Attachment Type

☒ Results

Upload Certification Package Fields

If you do choose *Acquire from Package* as your product data source, the other sections on the page will be replaced with the Upload Certification Package section. If you use the *Provide Manually* option, which is what we recommend for new users of the catalog, you will not see the Upload Certification Package section until after you have submitted your certification request.

- **File** - This is the results file you wish to upload to the catalog.
- **Description** - Enter a brief description of the results file here. We recommend something that will help you differentiate it from other results files submitted to this or other certification requests.

HARDWARE DETAILS

Vendor *

Mycompany

Make *



Ultraserver



Add New Make:

Model *

X16

Category *

Server

Product URL

http://www.mycompany.com/ultraservers.

Support URL

http://support.mycompany.com/ultraserve

Specification URL *

http://www.mycompany.com/ultraservers.



Production



Preproduction

Hardware Details Fields

- **Vendor** - This dropdown box holds the names of all the companies with previously certified systems. For our example we have used "Mycompany" as a hypothetical vendor. If your vendor name does not appear here, select **Unlisted Vendor** and the review team will add your company's name to the list.
- **Make** - This dropdown box holds the names of all the families of computers that have already been submitted for the vendor chosen above. It is the "Ultraserver" in the hypothetical "Mycompany Ultraserver X16". If a new make is needed, use the next field.
- **Add New Make** - Add a new system make here if it is not already in the **Make** dropdown box.
- **Model** - This is the model of computer being tested. It is the "X16" in "Mycompany Ultraserver X16".

- **Category** - This is the type of item you are certifying. The selection you make has no effect on the number or type of tests that are run, it simply indicates the device's intended role to perspective buyers. As the hypothetical "Mycompany Ultraserer X16" system is a server, the proper selection for this field is **Server**.
- **Product URL** - This optional field allows you to enter a different URL for the home of your product. For example, if you provide us with a link to a PDF document with system specs in the **Specification URL** section, you can use this field to link to the HTML page that describes all your system's or component's features.
- **Support URL** - This optional field allows you to enter a link to the support section on your website that handles questions about this system or component.
- **Specification URL** - This field is provided for you to enter the specifications URL for your product. The review team will use the information here to build the official list of tests, called the official test plan, that must be completed in order to certify the system. We discuss the official test plan in greater detail in a later section. If the URL cannot be included (the system is unreleased and has no public URL, for example), we allow you to attach document that contains the specs instead. You should use your main corporate website's URL as the link here if that is the case. We discuss attachment of specs documents in [Section 2.9, "Submitting Specs When no Public URL Exists"](#). If you chose to create a Hardware Specification entry at the same time as your hardware certification request, this specification information will be used to create the specification entry.
- **Production / Preproduction** - This radio button allows you to indicate whether the specs link is for a real shipping system or if it is preproduction only. If **preproduction** is selected, our review team will need to see final production specs before the certification can be completed.

CERTIFICATION DETAILS

Product

Red Hat Enterprise Linux

Version

RHEL 6

platform

x86_64

Certification Details Fields

- **Product** - This is the software you are certifying your hardware on. The only option at the moment is "Red Hat Enterprise Linux", but other products may be added in the future.
- **Version** - The major version number of Red Hat software product you are certifying with. For Red Hat Enterprise Linux, this will be 4, 5, 6, etc. The update release, if any, will be set using data from test results packages.

- **Platform** - The architecture you are certifying against. If you will be certifying multiple architectures, select any of the architectures you will be certifying with from this box. The others will be added automatically using data from test results packages.

UPLOAD SPECIFICATION

File

SELECT

Description

Describe the attachment briefly.

Attachment Type

☒ **Specification Document**

The Upload Specification section allows you to attach a specification document if a web-based version is not available. The specification document functionality is covered in [Section 2.9, “Submitting Specs When no Public URL Exists”](#).

Have you read the certification policy document?

- ☒ **Yes**, I'm ready for the next step.
- ☐ **No**, please let me read it --[certification policy document](#)

CONTINUE

The "Have you read the certification policy document" section confirms that you have read the [Policy Guide](#)², which is the official document that contains the requirements for certification. It was used to prepare this guide and it is the document that the review team uses when they build the official test plan for submitted hardware. The link in the **No** choice will take you to the document if you want to read it. Click the **Continue** button when you are ready to move to the next step.

² http://docs.redhat.com/docs/en-US/Red_Hat_Hardware_Certification/1/html-single/Policy_Guide/index.html

Certification Summary

Reporter	tester@mycompany.com
Vendor	Mycompany
Make	Ultrasever
Model	X16
Category	Server
URL	http://www.mycompany.com/ultraservers/spec
Product	Red Hat Enterprise Linux
Version	6
Kernel	Provided by Package
Platform	x86_64

The screen you are taken to after clicking the **Continue** button shows a summary of the information you have provided. If this is correct, proceed to the next section. Otherwise, click on the **Start Over** link at the bottom of the page.

Create Red Hat OpenStack Certification

☒ Yes ☐ No

If you wish to automatically create a Red Hat Enterprise Linux OpenStack Platform certification with your Red Hat Enterprise Linux 6 x86_64 certification, select the **Yes** option here. This option only appears for Red Hat Enterprise Linux OpenStack Platform partners.

Hardware Sample

Red Hat requires a representative sample of this hardware to be sent to its labs for support purposes under the terms of the Hardware Requirements agreement. Please acknowledge agreement with this requirement by clicking "Yes". If you have already sent similar hardware to Red Hat (e.g., identical chipset/CPU/BIOS/NIC), please click "Yes" and indicate the model that you have already sent to Red Hat and note any similarities and differences between this system and the model sent to Red Hat.

No ☒ Yes ☐

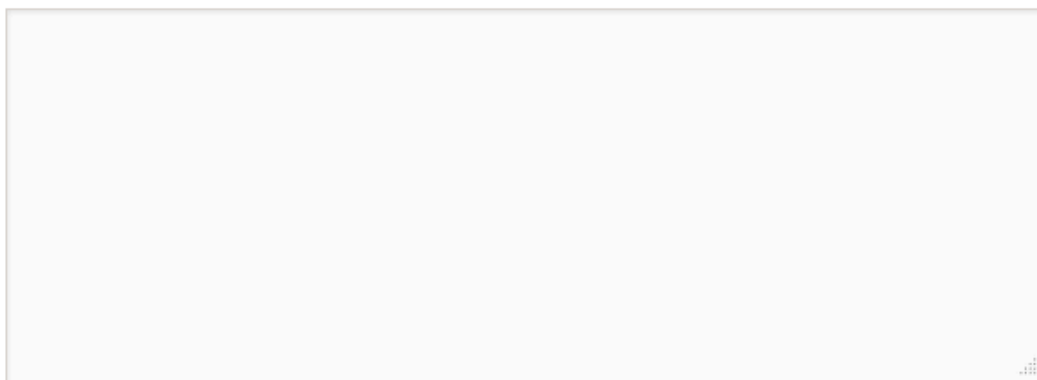
If yes, please enter shipment information below.

Shipped on SHIPMENT_DATE
Via Fedex tracking number 1234567890
to
Hardware Certification
Red Hat
314 Littleton Road
Westford, MA 01886

This section of the document provides an area for you to enter sample hardware shipment information. Red Hat requires a representative hardware sample for each unique model of hardware that we certify. If the information is not provided at this time, it will need to be provided before the certification can be published. This same section is available in the certification entry after it has been submitted so the information can be provided at any time.

Add Additional Comment

Please provide extra detail regarding any BIOS or Firmware settings and/or any testing experience comments.



Does this look correct?

If so click on **Commit**. If not, please back up and make the necessary corrections.

COMMIT

START OVER

If there is anything you wish to tell us about the certification you are submitting you can enter it in the **Additional Comment** box. It may also be left blank. Once you are satisfied that all the information is correct, click the **Commit** button to create the certification request.

At this point, the certification is open and ready for analysis by the review team. Move ahead to [Section 2.10, "How Red Hat Creates the Official Test Plan"](#) if you are performing a standard certification, or continue reading to learn about the other types of certifications.

2.3. NEW - Opening a Hardware Specification Entry

In November 2013, we introduced an update to the Red Hat Hardware Catalog that enabled the first phase of a feature we call the "hardware layer". This feature will eventually be used with all the different types of certifications and, when fully implemented, will allow us to track the hardware in each of the certifications that are submitted to the catalog. We will be able to use that information to shorten the certification process whenever the same hardware is certified on multiple versions of RHEL or on multiple Red Hat products. The piece of the hardware layer that has been implemented to date is called a "hardware specification entry". It is a separate hardware catalog entry that contains a list of all the testable components that make up a system or component, the category each component falls under, and the features the component provides. Hardware specification entries are never closed, and they are never visible to the general public on the hardware catalog.



Note

The use of hardware specification entries is completely optional at this time.

Hardware specification entries are typically opened with a new certification request. Directions can be found in [Section 2.2, "Opening a Standard Certification Request"](#) in the "Create New Entry Fields" explanation.

They can also be opened independently of any certification requests. To do this, follow the same directions from the previous link, but choose the option *Hardware Specification* from the dropdown box, and fill in the requested information. Note that there will be no **Certification Details** box to fill in if you are only creating a hardware specification entry.

After the entry is created, the review team will analyze the specifications URL or document that is associated with it, and populate the entry with testable components. An excerpt from an example hardware specification entry can be seen below:

Hardware Specification

NETWORK & STORAGE

Cloverleaf Industries Mergelane 1000	Integrated	1GIGETHER
Cloverleaf Industries Passinglane 10k	Optional	10GIGETHE

VIDEO

Acme Products Supervid 3k3	Optional	VIDEO
-----------------------------------	----------	-------

In the example you can see the vendor, make, and model information for each component, the category the component belongs to, and the tests required to certify the component. If you feel that there are problems with any of the components in the hardware specification entry, talk to your Red Hat support contact to resolve the issue.

2.4. NEW - Opening Standard Certification Requests from a Hardware Specification Entry

If you have created a hardware specification entry for a system or component, you should use it as a starting point for any standard hardware certification requests for that system. To create a new certification, open the hardware specification entry for your system, select the **Certifications** or **Show All** option in the menu, and scroll down to the **Create New Original Certification** section.

Create new Original Certification

CERTIFICATION INFORMATION*

Product *	Red Hat Enterprise Linux
Category *	Server
Version *	5
Platform *	x86_64
Create Red Hat	<input type="radio"/> Yes
OpenStack Certification	<input checked="" type="radio"/> No

CREATE ORIGINAL CERT

- **Product** - This is the software you are certifying your hardware on. The only option at the moment is "Red Hat Enterprise Linux", but other products may be added in the future.
- **Category** - This is the type of item you are certifying.
- **Version** - The major version number of Red Hat software product you are certifying with. For Red Hat Enterprise Linux, this will be 4, 5, 6, etc. The update release, if any, will be set using data from test results packages.
- **Platform** - The architecture you are certifying against. If you will be certifying multiple architectures, select any of the architectures you will be certifying with from this box. The others will be added automatically using data from test results packages.

- **Create Red Hat OpenStack Certification** - Automatically create a Red Hat Enterprise Linux OpenStack Platform certification entry for this new certification request. The option can only be set to "yes" when creating a Red Hat Enterprise Linux 6, x86_64 certification.

Click the **Create Original Cert** button when finished. The catalog will take you to the new certification request. You can see a link to the hardware specification entry in the "Relationships" section of the new certification. If you view the hardware specification entry again, you will find that the new certification request has been added to its "Certifications" section.

The process now moves to [Section 2.10, "How Red Hat Creates the Official Test Plan"](#).

2.5. Opening a Supplemental Certification Request

Certified systems often have component changes during their life cycles as new components become available. Any change to a certified system's specs that would require a retest due to the rules of the [Policy Guide](#)³ require the vendor to submit a special certification request called a supplemental certification. Supplemental certifications allow us to confirm that the changed components work with RHEL. There are two major differences between regular and supplemental certs:

- Supplemental certs only require the tests necessary to confirm the functionality of the new component(s). This means that most supplemental certs contain fewer tests than the original full system cert.
- Supplemental certs are not published when they are completed. The original certification entry for the system remains the only one viewable by the public (if it was set as published).

Processor upgrades are the most common type of supplemental certification, so we can use them in an example of the supplemental certification process. Acme Computers is updating their flagship workstation system with two new ten-core, 64-bit processors from Intel. Their original certification for this system used an earlier generation processor. We know from the [Policy Guide's hardware class requirements](#)⁴ section that a change in the processor generation requires retesting, so a supplemental cert is in order. This is the procedure that Acme's tester would follow to submit the supplemental cert for the upgraded processor:

1. The tester opens a new certification entry on the hardware catalog, using the procedures described in [Section 2.2, "Opening a Standard Certification Request"](#).
2. She uses the same vendor, make, and model as well as the same specs URL as the original cert entry and places a note in the new cert to explain that it is a supplemental cert, created to add new processor support to an existing certified model.
3. She makes sure to include a link to the existing certified system's hardware catalog entry in one of the comments.
4. The review team analyzes the specs at the included link and determines the differences between them and the original cert's official test plan. They create a new test plan to test these differences. In this case, the only difference they find is the new processor, so the review team creates a test plan containing the appropriate entries for each architecture.

³ <https://hardware.redhat.com/doc/index.html>

⁴ https://access.redhat.com/site/documentation/en-US/Red_Hat_Hardware_Certification/1/html-single/Policy_Guide/index.html#sect-Policy_Guide-Creating_the_Test_Plan-Hardware_Class_Requirements

5. The tester runs the requested tests on the same version and release of RHEL that were used for the original cert. If a newer update release of RHEL is used, a note is placed in the original certification indicating that support for the new components requires the new update release.
6. She submits the test results when completed.
7. The review team analyzes the results and gives credit for each passing test that matches the items in the test plan. Any missing tests would be noted and the tester would be asked to re-run them.
8. When all the tests have passed, the review team marks the entries as completed.

Then the review team closes the cert, leaving it private. Supplemental certs are not published because the information in them is for internal use only. The changes from the original system are reflected in the specs URL associated with the original cert, whose contents will have changed since the original certification was submitted.

2.6. NEW - Opening a Supplemental Certification Request from a Hardware Specification Entry

If you have created a hardware specification entry for a system, you should use it as a starting point for any supplemental certification requests for that system. To begin, open the entry for the system. If you do not know the ID for the entry, you can find it the **Relationships** section of any associated hardware certification requests. Here are the steps to create the supplemental certification request from the hardware specification entry:

1. Click the **Dialog** or **Show All** menu and find the **Additional Comment** text box. Add a comment requesting a supplemental certification for new hardware and provide a link to the updated specs if the current URL or specs document is no longer valid. Set the comment to "Need additional information from: Reviewer" and click the **Save Changes** button when finished.
2. The review team will analyze the new hardware and add it to the hardware contained in the specification entry. They will place an update in the specification entry when they are finished, letting you know that you can create the new supplemental certification request.
3. Open the hardware specification entry and click the **Certifications** or **Show All** menu and scroll down to the "Create New Supplemental Certification" section. Note that this will not be visible until the review team has added supplemental hardware to the specification entry:

Create new Supplemental Certification

Create new supplemental certification for

Select a Certification

▼

CREATE SUPPLEMENTAL CERT

4. Select the proper certification in need of a supplemental entry from the dropdown box, and click the **Create Supplemental Cert** button when finished. You will be taken to the new supplemental certification request.
5. The review team will create a test plan in the new supplemental certification request based on the supplemental items they just added to the hardware specification entry.
6. You should run the requested tests on the specified hardware and submit results to the new supplemental certification request.
7. The review team will close the supplemental certification request, leaving it unpublished, once passing results have been received.

2.7. Opening a System Pass-Through Certification Request

System pass-through certifications essentially create copies of certified systems, listing them under a different vendor name, a different make, or a different model. They are used in several different scenarios:

1. A vendor sells their systems to partners who rebrand and resell them to the public. The reseller makes no more than minor (see note below) changes to the system, packages it and sells it under their own name.
2. Two or more systems are being certified by the same vendor, with one system being a superset of all the others. The superset of the two or more systems can be certified, and pass-through used to create entries for all the systems that are subsets of the main system without additional testing.

Please talk with your Red Hat Support contact to determine if pass-through certification will work for your specific system(s).



Note

Minor changes as described here are things such as new splash screens or the addition of further optional hardware components. Major changes are things like differences in integrated hardware or BIOS feature additions, which would make a system or component ineligible for the pass-through program. Your Red Hat Support contact can help you determine if your items are eligible for pass-through.

System pass-through certifications are begun from the original system certification entry, which is referred to as the "parent cert" for purposes of pass-through. Scroll to the top of the parent cert, click on the **Advanced** menu, and find the **Create New Pass-Through Certification** box.

Create new Pass-Through Certification

Vendor	Mycompany (edit)
Make	Ultraserver (edit)
Model	<input type="text" value="X15"/>
URL	<input type="text" value="http://www.mycompany.com/specs/ult"/>

Have you read the certification policy document?

☐ Yes, I have read the [Pass-Through policies](#) and confirm this model is a subset and I understand the responsibilities that accompany pass-through certifications.

CREATE CLONE

Fill out the information there as follows:

- **Vendor** - The name of the company selling the newly created pass-through system. This could be a reseller in the case of a rebranded system, or it could be the original company if the pass-through cert is being used to handle a series of systems which are a subset of the parent and are being sold by the original company.
- **Make** - The make assigned to the pass-through system. For our example, we are using the Mycompany Ultraserver X16 as a superset of a new model X15 that is also made by Mycompany. In this case, the make would stay "Ultraserver"
- **Model** - The model name for the pass-through system. The new pass-through system is the Mycompany Ultraserver X15, so the model here is "X15".
- **URL** - The vendor's specs URL for the rebranded or subset system. We must have this in order to analyze any differences between the parent and the clone.

Check the checkbox after you have read the pass-through policy guide (a link to the guide is displayed on the pass-through page) to create the new cert. The review team will analyze the pass-through certification request's specs URL and notify you if any differences between it and the original system's specs are found. If none are found, they will approve the pass-through cert without additional testing. If differences are found, they will either reject the request with an explanation if the systems have major differences, or request supplemental certification (see [Section 2.5, "Opening a Supplemental Certification Request"](#)) of the additional hardware to fill in any gaps. The supplemental cert items will be added to the pass-through cert request's test plan and will be handled within the pass-through cert. When any supplemental items are completed with passing results, the pass-through cert will be approved.

2.8. NEW - Opening a System Pass-Through Certification Request from a Hardware Specification Entry

As mentioned in the previous section, system pass-through certifications essentially create copies of certified systems, listing them under a different vendor name, a different make, or a different model.

If you have created a hardware specification entry for a system, you should use it as a starting point for any pass-through certification requests based on that system. To begin, open the entry for the system. If you do not know the ID for the entry, you can find it the **Relationships** section of any associated hardware certification requests. When the entry is open, click the **Pass-Through** menu or the **Show All** menu, and find the **Create New Pass-Through System** box.

CREATE NEW PASS-THROUGH SYSTEM

Vendor *

Mycompany

Make *

Ultrasever

Model *

X15

Category

Server

Product Home Page URL

http://www.mycompany.com/ultraservers/X15

Support Home Page URL

http://support.mycompany.com/ultraservers/

Specification URL

http://www.mycompany.com/ultraservers/sp

☒ **Production** ☐ **Preproduction**

Have you read the certification policy document

☐ **Yes**, I have read the [Pass-Through policies](#) and confirm this model is a subse
understand the responsibilities that accompany pass-through certifications.

CREATE PASS THROUGH SYSTEM

Fill out the information there as follows:

- **Vendor** - The name of the company selling the newly created pass-through system. This could be a reseller in the case of a rebranded system, or it could be the original company if the pass-through cert is being used to handle a series of systems which are a subset of the parent and are being sold by the original company.
- **Make** - The make assigned to the pass-through system. For our example, we are using the Mycompany Ultraserver X16 as a superset of a new model, the X15, that is also made by Mycompany. In this case, the make would stay "Ultraserver"
- **Model** - The model name for the pass-through system. The new pass-through system is the Mycompany Ultraserver X15, so the model here is "X15".
- **Category** - The type of item you are certifying. The X15 is another server, so the category should be set to "Server".
- **Product Home Page URL** - This optional field allows you to enter a different URL for the home of the pass-through product.
- **Support Home Page URL** - This optional field allows you to enter a link to the support section on the pass-through system vendor's website that handles questions about the pass-through system or component.
- **Specification URL** - The vendor's specs URL for the rebranded or subset system. We must have this in order to analyze any differences between the parent and the clone.
- **Production / Preproduction** - This radio button allows you to indicate whether the specs link is for a real shipping system or if it is preproduction only. If **preproduction** is selected, our review team will need to see final production specs before the certification can be completed.

Check the checkbox after you have read the pass-through policy guide (a link to the guide is displayed on the pass-through page) and click the **Create Pass Through System** to create the new cert. The review team will analyze the pass-through certification request's specs URL and notify you if any differences between it and the original system's specs are found. If none are found, they will approve the pass-through cert without additional testing. If differences are found, they will either reject the request with an explanation if the systems have major differences, or request supplemental certification (see [Section 2.5, "Opening a Supplemental Certification Request"](#)) of the additional hardware to fill in any gaps. The supplemental cert items will be added to the pass-through cert request's test plan and will be handled within the pass-through cert. When any supplemental items are completed with passing results, the pass-through cert will be approved.

2.9. Submitting Specs When no Public URL Exists

There may be times when a public URL is unavailable for Red Hat's review team to use when creating the official test plan. In this situation, we ask that you either supply the specifications of the hardware in an update (use the **Add Additional Comment** box) or upload a document that contains this information. You can attach a specs document by using the **Upload** section.

UPLOAD RESULTS PACKAGE OR SPECIFICATION

File

SELECT FILE

*Currently supported test suite versions: **hwcert-dient-1.6.4-57**, v7-1.6.4-22. Currently acceptable maximum test result file size: **16MB***

Description

Describe the attachment briefly.

Comment

(optional) Add a comment about this attachment to the certification.

Attachment Type

☐ Results

☐ Specification Document

Simply click in the **File** field and find the document on your hard drive. Specify the purpose of the document in the **Description** field, and add any additional information you wish in the **Comment** field. Then change the Attachment Type to **Specification Document** and click the **Select File** button.

Please note that we will need a public specs document before the certification is released, as we want to ensure that the specs provided to us match the specs provided to the customer. Please

contact your certification representative if no public document will be available as this may affect the certification status.

2.10. How Red Hat Creates the Official Test Plan

The next step in the process is the creation of the **official test plan**, which is the list of hardware that must be tested in order for a system to achieve certification. Our review team reads through the specifications URL, specs document, or list of specs you provide to us to determine which hardware should be considered part of the system model. They do this by categorizing the listed hardware into three different groups:

- **Integrated** - Hardware that is part of the motherboard and is always present in the system. Integrated hardware must be tested as part of the certification process, so it will appear on the official test plan.
- **Optional** - Hardware that can be ordered along with the model and is considered to be part of it, based on the intended function of the model and on how the hardware is presented in the specs URL. Examples of optional hardware can include NICs, video cards, FC cards, etc. Optional hardware must be tested as part of the certification process, so it will appear on the official test plan. Note, optional hardware was formerly known as "Config to Order" or "CTO" hardware.
- **Additional** - Hardware that can be ordered along with the model but is distinct from it, based on the intended function of the model and how the hardware is presented in the specs URL. An example would be a digital camera or a printer on a standard desktop system. Testing is not required for additional hardware, so it does not appear on the official test plan. Note, additional hardware was formerly known as "Aftermarket" or "AM" hardware.

These three designations are usually clear, but there can be confusion between the optional and additional classifications. Our review team looks at several aspects of the hardware to help make their decision. One attribute they check is the intended purpose of the system. For example, if a submitted system is a standard desktop model, a printer would be considered "additional" because printing is not a clear part of the system's core function. If, however, the computer is a desktop model that is sold as a check printing solution, the printer would be classified "optional" and testing would be required as the purpose of the machine is to print checks.

The team also observes the presentation of hardware in the specs / ordering URL. Hardware must be organized in such a way that it is clear to Red Hat and to customers that the hardware is optional or that it is additional. If all the available parts for a specific system are listed the same way on the same page, we would consider all the items to be optional items and require them to be tested. An example would be:

Chipset	Intel® Q35 Express Chipset (Non vPro)
Processor	Intel® Core 2 Duo E8400 Processor (3 GHz, 6 MB L2 Cache, 1333 MHz) or Higher Intel® Core 2 Duo E7300 Processor (2.66 GHz, 3 MB L2 Cache, 1066 MHz) or Higher Intel® Pentium Dual Core E5200 Processor (2.5 GHz, 2 MB L2 Cache, 800 MHz) or Higher Intel® Pentium Dual Core E2200 Processor (2.2 GHz, 1 MB L2 Cache, 800 MHz) or Higher Intel® Celeron Dual Core E1200 Processor (1.6 GHz, 512 KB L2 Cache, 800 MHz) or Higher Intel® Celeron 430 Processor (1.8 GHz) 512 MB L2 Cache, 800 MHz or Higher
System Memory	Single/Dual Channel DDR2 RAM 800 MHz (PC2 6400)/667 MHz (PC2 5300) expandable to 8 GB, 4 DIMM Slots
Video Graphics	Integrated Intel Graphics Media Accelerator 3100 support up to 256 MB Video Memory through DVMT
Hard disk drive	80 GB or Higher SATA 3.0Gbps, 7200 RPM, Integrated 3 SATA Ports
Floppy disk drive	1.44 MB Floppy disk drive (Optional)
Other devices	Choice of 52X Internal CD-ROM drive, 16X Internal DVD ROM, 16X DVD 52X32X52 Internal DVD Combo drive, 18X Internal DVD Writer Drive
Network controller	Integrated 10/100/1000 Mbps Ethernet controller with WOL support

If one or more of the options were set aside in a separate "Also Available", "Optional Hardware" or "Sold Separately" section, these designations would indicate to us that the hardware is considered to be an accessory rather than part of the model, so we would classify it as "additional" and not require testing. Here is an example of an additional hardware section from an IBM xREF specs page:

512MB PC2-5300 ECC DDR2 667MHz	41Y2726
2GB(2x1GB) PC2-5300 ECC DDR2 667MHz	41Y2729
4GB(2x2GB) PC2-5300 ECC DDR2 667MHz	41Y2732
IBM 160GB SATA (7.2K) simple-swap HDD	39M4504
IBM 250GB SATA (7.2K) simple-swap HDD	39M4508
IBM 500GB SATA (7.2K) simple-swap HDD	39M4514
IBM 750GB SATA (7.2K) simple-swap HDD	43W7572
160GB Hot-Swap SATA HDD	39M4522
250GB Hot-Swap SATA HDD	39M4526
500GB Hot-Swap SATA HDD	39M4530
750GB Hot-Swap SATA HDD	43W7576
73GB 15K 3.5" SAS Hot-Swap HDD	40K1043
146GB 15K 3.5" SAS Hot-Swap HDD	40K1044
IBM HH SATA Multiburner Comb drive (DVD-RAM)	43W8265
IBM Remote Supervisor Adapter II slimline	39Y9566
IBM 5Ux20D Tower-to-Rack Kit	42C8923
IBM SAS HBA Controller	25R8060

Other options at ibm.com/products/us

The labeling of the section does not have to match our examples, but it should be clear to everyone that this hardware is separate from the model. Please keep in mind that you cannot claim "additional" status for hardware just to exempt it from testing. It must clearly be "extra" to the system (not required to perform the intended task of the system) and it must be presented in a way that clearly indicates its additional status.

Another status that affects testing requirements is unique functionality. If a piece of optional hardware does not provide a unique function, it can be exempted from testing. For example, in the previous section we talked about a list of network cards that was considered optional. If one of those cards did not have a driver in Red Hat Enterprise Linux, as long as there is another card in the list that provides exactly the same functionality, that card does not need to be tested. In this situation we ask for vendors to clearly label cards with the operating systems that they do support, rather than listing them with those that are unsupported.

Whether or not the hardware is of a supported class also affects its inclusion in the test plan. There are only a few types of hardware, like Firewire controllers, that fall into this "unsupported class of hardware" category. Firewire is not a supported type of hardware in Red Hat Enterprise Linux so it would not be required to be tested.

After the hardware has been categorized, the integrated and optional hardware is put into the test plan. An excerpt from a newly created test plan can be seen below:

64bit Requirements		
CDROM SATA DVD+/-RW	Confirmed	<input type="checkbox"/>
<div>-----</div>		
CORE 2x Intel Xeon 5500 series processors(X5570/X5550/E5540)	Confirmed	<input type="checkbox"/>
<div>-----</div>		
DVD SATA DVD+/-RW	Confirmed	<input type="checkbox"/>
<div>-----</div>		
MEMORY 16 GB	Confirmed	<input type="checkbox"/>
<div>-----</div>		
NETWORK Broadcom NetXtreme II 5709 Quad Port Gigabit Ethernet PCIe Adapter Card with TOE and iSCSI HBA	Confirmed	<input type="checkbox"/>
<div>-----</div>		
NETWORK Broadcom NetXtreme II 57711 Dual Port 10 Gigabit Ethernet PCIe Adapter Card with TOE and iSCSI HBA	Confirmed	<input type="checkbox"/>
<div>-----</div>		
USB 2 Ports (2x front, 2x rear)	Confirmed	<input type="checkbox"/>
<div>-----</div>		
VIDEO Matrox G200 core embedded into the ServerEngines Pilot-2 BMC	Confirmed	<input type="checkbox"/>
<div>-----</div>		

You can see that this portion of the plan is for 64-bit Red Hat Enterprise Linux and that each line contains the following information:

- Test name and any descriptive information for the hardware that is covered by the test.
- "Confirmed" checkbox that is checked by the review team when the test is marked as passing.
- A drop-down box that identifies which run of the test contains the passing result that should be used for certification purposes. The box contains a dashed line when no passing results file has been selected by the review team. This could be because no passing result has been achieved on the tests that have been submitted, the review team has not yet processed the the results package that contains the passing results, or no results have yet been submitted for this item.

The dropdown box also contains a selection for leveraging of other test results, which is discussed in the section entitled [Section 2.11, "Leveraging of Existing Test Results"](#).

2.11. Leveraging of Existing Test Results

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 company or its agent* has performed.

Hardware that is eligible for leveraging consists of:

- Audio cards (non-integrated)
- Optical drives (CD/DVD/Blu-Ray)
- Storage controllers (non-integrated)
- Storage devices (Hard drives, SATA/SAS SSD drives)
- Network cards (non-integrated)
- Video cards (non-integrated)

- Tape drives (non-integrated)

If you would like to use leveraging to cover the testing of specific components, place a comment in the certification telling the review team which hardware you would like to leverage and include a link to the certified system that contains the passing results for the hardware. The review team will fill out the test plan with a leverage section and point to the test from the other certification entry. You can see an example of leveraging in the test plan in [Section 3.8, “Monitoring Certification Progress”](#).

Now you know what you need to test, but you will need to set up your test environment before you can begin. We will talk about that in the next section.

2.12. Preparing the Test Environment

Before you can begin running the tests listed in the official test plan, you will need to set up a test environment. Here is what you will need:

1. RHEL ISO files in the architectures that you will be certifying on
2. The MRG Realtime ISO for RHEL 6, x86_64 and a dependency file (**libxslt-python**) if you want to certify for Red hat Enterprise Linux MRG Realtime (this is optional)
3. The **hwcert-server** package (RHEL 6 only), the **hwcert-info** package, the **hwcert-client** package and its dependencies of **dt**, **lmbench** and **stress** in the architectures that you will be certifying on
4. The **kernel-debuginfo** and **kernel-debuginfo-common** packages required by your specific version and architecture of RHEL:

RHEL 5 - 32-bit

- **kernel-debuginfo**
- **kernel-debuginfo-common**
- **kernel-xen-debuginfo**
- **kernel-PAE-debuginfo**

RHEL 5 - 64-bit

- **kernel-debuginfo**
- **kernel-debuginfo-common**
- **kernel-xen-debuginfo**

RHEL 6 - 32-bit

- **kernel-debuginfo**
- **kernel-debuginfo-common-i686**

RHEL 6 - 64-bit

- **kernel-debuginfo**
- **kernel-debuginfo-common-x86_64**

Optional RHEL 6 - 64-bit MRG Realtime kernel

- **kernel-rt-debuginfo**
- **kernel-rt-debuginfo-common-x86_64**

5. A web server that will host the following items:
 - a. RHEL install trees
 - b. RHEL kickstart files
 - c. **hwcert** tool yum repositories
 - d. **kernel-debuginfo** yum repositories
 - e. Optional MRG Realtime yum repositories (install tree, debuginfo, dependencies)
6. A system that will function as a test server for the network and kdump tests
7. Additional support equipment for certain tests (see [Section A.1, “Test Procedures”](#) for more information) including:
 - a. Appropriate optical media for any Blu-Ray/DVD/CD drive tests
 - b. A USB device, such as a mouse, to test the function of any free USB ports
 - c. Speakers (if none are built in) and a microphone (if none is built in) or patch cable to test the function of any sound cards
 - d. Expresscards with both USB and PCIe interfaces (this can be one card with both interface types or two cards that do one interface each) for any Expresscard slots
 - e. A PC card (formerly called PCMCIA card) to test the function of any PC card slots
 - f. Blank tape for any tape drives
8. Systems in need of certification, which we refer to as Systems Under Test or SUTs.

We will cover items one through five in this section.

The RHEL ISO(s), hwcert packages and all their dependencies, and the kernel debuginfo packages can be downloaded from the Red Hat Customer Portal (<http://access.redhat.com>). Log in to the portal using your username and password, then follow the steps below to get all the ISO files and RPM packages you need.

1. Download the RHEL DVD ISO file: After logging in to the portal, follow the shortcut links below to download the appropriate binary DVD ISOs, or follow the directions in the [Red Hat Enterprise Linux 6 Installation Guide](#)⁵ to find the RHEL 5 or RHEL 6 ISOs and download them manually. Store these files in a unique, web-accessible location on your web server (see step five for more details).

RHEL 5 ISO Download Links

- 32-bit x86: <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=6949>
- 64-bit IBM POWER: <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=6972>

⁵ https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html-single/Installation_Guide/index.html#ch-Obtaining_Red_Hat_Enterprise_Linux

- 64-bit IBM System Z: <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=6972>⁶
- 64-bit x86_64: <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=6972>⁷

RHEL 6 ISO Download Links

- 32-bit x86: <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=10358>
- 64-bit IBM POWER: <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=10417>
- 64-bit IBM System Z: <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=10438>
- 64-bit x86_64: <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=10486>



Note

These links will always take you to the current update release for the version and architecture of RHEL that you selected. If you wish to certify on an older update release, click the **View ISO Images for Older Releases** link at the bottom of the page, choose the link to the older update release of RHEL you need, such as **Red Hat Enterprise Linux 6.1 Server (AMD64/Intel EM64T)** and download that binary ISO instead.

2. Optionally, download the MRG Realtime ISO file and other dependencies. Store these files in a unique, web-accessible location on your web server (see step five for more details):
 - a. After logging in to the portal, follow the shortcut link below to download the appropriate MRG Realtime packages, or follow this procedure to find them manually. Click the **Downloads** link in the portal's menu bar. When the next page loads, scroll down to the Red Hat Enterprise Linux graphic and click the **Download Software**. This will take you to the RHEL ISO download page. Find the RHEL 6, x86_64 link and click the plus sign next to it to expand the list of child channels. Click the **MRG Realtime v. 2 (for RHEL 6 Server x86_64)** link and download the binary ISO file(s).

Red Hat Enterprise MRG Realtime 2.x Download Link

- <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=12041>
- b. Download the MRG Realtime dependency package, **libxslt-python**: After logging in to the portal, follow the shortcut link below:

libxslt-python Package Download Link

- <https://rhn.redhat.com/rhn/software/packages/details/Overview.do?pid=726827>

⁶ <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=6976>

⁷ <https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=6957>

Click the **New Versions** tab on the page that appears and check for new versions of the package. If none exist, click the **Details** tab to go back. If entries do appear, find the newest one that is associated with the channel "RHEL Server Optional (v. 6 64-bit x86_64)" and click on its name. These steps will put you on the download page for the newest version of the package. Now, scroll to the bottom of the page and click the **Download Package** link to download **libxslt-python**.

3. Download the hwcert packages and their dependencies: Follow the link(s) below that matches the version(s) and architecture(s) of RHEL you will be certifying and download the the latest versions of the packages **hwcert-server** (RHEL 6 only), **hwcert-info**, **hwcert-client**, **dt**, **lmbench**, and **stress**. Store these files in a unique, web-accessible location on your web server (see step five for more details).

RHEL 5 hwcert Package Download Links

- 32-bit x86: <https://rhn.redhat.com/rhn/channels/PackageList.do?cid=6955>
- 64-bit IBM POWER: <https://rhn.redhat.com/rhn/channels/PackageList.do?cid=6974>
- 64-bit IBM System Z: <https://rhn.redhat.com/rhn/channels/PackageList.do?cid=6978>
- 64-bit x86_64: <https://rhn.redhat.com/rhn/channels/PackageList.do?cid=6963>

RHEL 6 hwcert Package Download Links

- 32-bit x86: <https://rhn.redhat.com/rhn/channels/PackageList.do?cid=11131>
 - 64-bit IBM POWER: <https://rhn.redhat.com/rhn/channels/PackageList.do?cid=11134>
 - 64-bit IBM System Z: <https://rhn.redhat.com/rhn/channels/PackageList.do?cid=11137>
 - 64-bit x86_64: <https://rhn.redhat.com/rhn/channels/PackageList.do?cid=11140>
4. Download the debuginfo packages for the version and architectures of RHEL you are certifying. Store these files in a unique, web-accessible location on your web server (see step five for more details):
 - a. If you are certifying on RHEL 6, download the debuginfo packages for RHEL 6. This process will need to be performed once per architecture:

First, find the kernel file(s) on your DVD ISO or install tree. On RHEL 6, there will only be one kernel file included in each architecture (name starts with "kernel-"). For example, on RHEL 6.2, x86_64 arch, the name of the kernel RPM file is "kernel-2.6.32-220.el6.x86_64.rpm".

Next, click the **Downloads** link in the menu bar of the customer portal's main page (<http://access.redhat.com>). When the next page loads, click the **Packages** link. This will take you to the package search and download page. Enter the entire name of the kernel RPM file, extension and all, in the search box, choose **Name only** in the **What to search** section, then click the **In the following architectures:** radio button and select the architecture that matches the architecture listed in the kernel file's name. Note: "IA-32" is another name for "32-bit" or "x86". Click the **Search** button when ready.

You will be presented with a list of search results. These are the packages you need, sorted by version and architecture:

RHEL 6 - 32-bit

- kernel-debuginfo-\$VERSION
- kernel-debuginfo-common-i686-\$VERSION

RHEL 6 - 64-bit

- kernel-debuginfo-\$VERSION
- kernel-debuginfo-common-x86_64-\$VERSION

Click the names of the appropriate packages in the search results, then click the **Download Package** link to retrieve them. We recommend opening each in a new tab so that you do not have to use the **Back** button in your browser or repeat the search query. After retrieving the packages for your first architecture, you will need to repeat this step for any additional architectures on RHEL 6.

- b. If you are certifying on RHEL 5, download the debuginfo packages for RHEL 5. This process will need to be performed once per architecture:

First, find the "plain" kernel file on your DVD ISO or install tree. On RHEL 5, 32-bit, there will be three kernel files (names start with "kernel-", "kernel-xen-" and "kernel-PAE-"). On RHEL 5, 64-bit, there will be two kernel files (names start with "kernel-" and "kernel-xen-"). Here are some examples:

RHEL 5.8, 32-bit (IA-32)

- kernel-2.6.18-308.el5.i686.rpm
- kernel-xen-2.6.18-308.el5.i686.rpm
- kernel-PAE-2.6.18-308.el5.i686.rpm

RHEL 5.8, 64-bit (x86_64)

- kernel-2.6.18-308.el5.x86_64.rpm
- kernel-xen-2.6.18-308.el5.x86_64.rpm

We are interested in the "kernel-\$VERSION" file, like "kernel-2.6.18-308.el5.i686.rpm", for this process. We can use that filename to find all of the packages we need in one search.

Next, click the **Downloads** link in the menu bar of the customer portal's main page (<http://access.redhat.com>). When the next page loads, click the **Packages** link. This will take you to the package search and download page. Enter the entire name of the "plain" kernel RPM file, extension and all, in the search box, choose **Name only** in the **What to search** section, then click the **In the following architectures:** radio button and select the architecture that matches the architecture listed in the kernel file's name. Note: "IA-32" is another name for "32-bit" or "x86". Click the **Search** button when ready.

You will be presented with a list of search results. These are the packages you need, sorted by version and architecture:

RHEL 5 - 32-bit

- kernel-debuginfo-\$VERSION
- kernel-debuginfo-common-\$VERSION
- kernel-xen-debuginfo-\$VERSION

- kernel-PAE-debuginfo-\$VERSION

RHEL 5 - 64-bit

- kernel-debuginfo-\$VERSION
- kernel-debuginfo-common-\$VERSION
- kernel-xen-debuginfo-\$VERSION

Click the names of the appropriate packages in the search results, then click the **Download Package** link to retrieve them. We recommend opening each in a new tab so that you do not have to use the **Back** button in your browser or repeat the search query. After retrieving the packages for your first architecture, you will need to repeat this step for any additional architectures on RHEL 5.

- c. If you would like to complete the optional MRG-RT certification on 64-bit RHEL 6, download the debuginfo packages for the Realtime (RT) kernel. RT only works on the 64-bit architecture of RHEL 6, so there is no need to repeat for multiple architectures.

First, find the kernel-rt file on your MRG-RT ISO or MRG-RT install tree. For example, on MRG-RT 2.1, the name of the kernel-rt RPM file is **kernel-rt-3.0.25-rt44.57.el6rt.x86_64.rpm**.

Next, click the **Downloads** link in the menu bar of the customer portal's main page (<http://access.redhat.com>). When the next page loads, click the **Packages** link. This will take you to the package search and download page. Enter "kernel-rt" in the search box, choose **Name only** in the **What to search** section, then click the **In the following architectures:** radio button and select "x86_64". Click the **Search** button when ready.

You will be presented with a list of package names. Click on the "kernel-rt-debuginfo" and "kernel-rt-debuginfo-common-x86_64" packages, locate the versions that match the "kernel-rt" kernel present in your MRG-RT ISO or install tree and download them to your server. We recommend opening each in a new tab so that you do not have to use the **Back** button in your browser or repeat the search query.

5. Set up a web server to host the RHEL install trees, kickstart files and the hwcert files. This does not need to be a dedicated machine, it can be any web server with sufficient disk space to hold the RHEL install trees that you will be certifying (approximately 3.5 GB per architecture) with as well as the kickstart files and hwcert / debuginfo repositories (about 100 MB). It should *not* be the same machine as the network and kdump test server due to network capacity requirements (a system serving RHEL bits during an install will not have sufficient bandwidth to function as the network test server).
 - a. To set up the RHEL install tree, copy the entire contents of the RHEL ISO to a web-accessible directory on your web server. Be sure to use a naming scheme for your directories that takes architectures into account if you will be certifying multiple architectures. You can either loopback-mount the ISO file you downloaded (**mount -o loop filename.iso /directory/to/mount/iso/file**) and copy the files from there, or you can use a burned DVD and copy the files from that.
 - b. Copy the optional MRG Realtime files from the ISO you downloaded into a unique, web-accessible directory on your web server.
 - c. Next, copy the kickstart files to their location on the web server. You should use a directory naming scheme to allow for multiple versions and architectures of RHEL. You can read more

about creating and editing kickstart files in [Section 2.13, “Preparing a Kickstart File to Install RHEL 5 and hwcert-client”](#) and [Section 2.14, “Preparing a Kickstart File to Install RHEL 6 and hwcert-client”](#) of the guide.

- d. Copy **hwcert-server** (RHEL 6 only), **hwcert-info**, **hwcert-client** and its dependencies, **dt**, **Imbench**, and **stress** to a dedicated directory on the web server. Each architecture will need its own repository, so copy the files into directories named for the file architecture. You can re-use the hwcert repository for different update releases of RHEL, i.e. RHEL 6.3, x86_64 and RHEL 6.0, x86_64 will use the same hwcert repository.

Use the command **createrepo -p** . inside each of those directories to build a yum repository from the files. Do not forget the period at the end of the command as that indicates where the repo metadata files are to be stored (the current directory).

- e. Copy the optional MRG Realtime **libxslt-python** dependency package into a unique location on the web server. Use the same **createrepo -p** . command as in the last paragraph to create a yum repository out of the files in the debuginfo directory.
- f. Finally, copy the debuginfo packages to their location on the web server. Use the same **createrepo -p** . command as in the last paragraph to create a yum repository out of the files in the debuginfo directory. If you decided to run the optional MRG-RT certification you can put the kernel-rt debuginfo packages in the same directory / yum repo as the RHEL 6 x86_64 kernel debuginfo packages.



Important

The **createrepo** command creates different styles of yum repositories in RHEL 5 and RHEL 6. It is recommended that you run your **createrepo** commands from a RHEL 5 system for any RHEL 5 yum repositories and from a RHEL 6 system for any RHEL 6 yum repositories.

Here is one way you could lay out all the directories on your web server to serve all the necessary files to handle 32-bit and x86_64 64-bit RHEL certification. In this example, we are setting up an environment to certify RHEL 5.8 and 5.9, 32-bit and 64-bit (x86_64); RHEL 6.3 and 6.4, 32-bit and 64-bit (x86_64); and MRG Realtime 2.x:

- /var/www/html/rhel5/5.8/32 and /64 - The RHEL 5.8 install trees copied from the ISO or other location
- /var/www/html/rhel5/5.9/32 and /64 - The RHEL 5.9 install trees copied from the ISO or other location
- /var/www/html/rhel5/ks - Individual kickstart files capable of installing the update release(s) of RHEL 5 on which you will certify your systems
- /var/www/html/rhel5/hwcert-32 - Yum repository containing hwcert-client, hwcert-info, Imbench, dt and stress; These files can be used by all update releases of RHEL5 with arch i386/i686
- /var/www/html/rhel5/hwcert-64 - Yum repository containing hwcert-client, hwcert-info, Imbench, dt and stress; These files can be used by all update releases of RHEL5 with arch x86_64
- /var/www/html/rhel5/rhel5.8-32-debug - Yum repository containing debuginfo packages for the RHEL 5.8 32-bit kernel

- `/var/www/html/rhel5/rhel5.8-64-debug` - Yum repository containing debuginfo packages for the RHEL 5.8 x86_64 kernel
- `/var/www/html/rhel5/rhel5.9-32-debug` - Yum repository containing debuginfo packages for the RHEL 5.9 32-bit kernel
- `/var/www/html/rhel5/rhel5.9-64-debug` - Yum repository containing debuginfo packages for the RHEL 5.9 x86_64 kernel
- `/var/www/html/rhel6/6.3/32` and `/64` - The RHEL 6.3 install trees copied from the ISO or other location
- `/var/www/html/rhel6/6.4/32` and `/64` - The RHEL 6.4 install trees copied from the ISO or other location
- `/var/www/html/rhel6/ks` - Individual kickstart files capable of installing the update release(s) of RHEL 6 on which you will certify your systems
- `/var/www/html/rhel6/hwcert-32` - Yum repository containing hwcert-client, hwcert-server, hwcert-info, lmbench, dt and stress; These files can be used by all update releases of RHEL6 with arch i386/i686
- `/var/www/html/rhel6/hwcert-64` - Yum repository containing hwcert-client, hwcert-server, hwcert-info, lmbench, dt and stress; These files can be used by all update releases of RHEL6 with arch x86_64
- `/var/www/html/rhel6/rhel6.3-32-debug` - Yum repository containing debuginfo packages for the RHEL 6.3 32-bit kernel
- `/var/www/html/rhel6/rhel6.3-64-debug` - Yum repository containing debuginfo packages for the RHEL 6.3 x86_64 kernel and optionally, the debuginfo packages for the MRG Realtime kernel-rt package
- `/var/www/html/rhel6/rhel6.4-32-debug` - Yum repository containing debuginfo packages for the RHEL 6.4 32-bit kernel
- `/var/www/html/rhel6/rhel6.4-64-debug` - Yum repository containing debuginfo packages for the RHEL 6.4 x86_64 kernel and optionally, the debuginfo packages for the MRG Realtime kernel-rt package
- `/var/www/html/realtime/2.1` - Yum repository containing optional MRG Realtime packages
- `/var/www/html/realtime/dependencies` - Yum repository containing libxslt-python dependency for optional MRG Realtime testing

In a later section, we will perform an automated install of RHEL and hwcert on the network test server using the web server and kickstart files we created in this section.

2.13. Preparing a Kickstart File to Install RHEL 5 and hwcert-client

The simplest way to install RHEL 5 and the hwcert packages is by using a kickstart file. Kickstart files are text files containing instructions that automate the installation of RHEL and other RPM packages on a system. We have prepared example kickstart files for RHEL 5 i386 and x86_64 hardware certification environments and made them available here on our FTP server:

- <ftp://partners.redhat.com/a166eabc5cf5df158922f9b06e5e7b21/hwcert/kickstarts/>

The files of interest are **rhel5-x86_64-client.ks** (RHEL 5, x86_64 architecture) and **rhel5-i386-client.ks** (RHEL 5, i386 architecture), which install a RHEL 5 system that is ready to undergo certification. These files can be duplicated, renamed, and edited to enable you to install different update releases of RHEL 5 (rhel5.4-x86_64-client.ks, rhel5.5-x86_64-client.ks, rhel5.6-x86_64-client.ks, and so on) in your test environment.

You can learn more about kickstart files in Chapter 31 of the [Red Hat Enterprise Linux 5 Installation Guide](#)⁸, entitled [Kickstart Installations](#)⁹. To use a kickstart file to install your RHEL 5 SUTs, follow the directions in the Installation Guide referenced above to prepare a kickstart server, customize the example files to match your environment and then copy them to your server. The kickstart files we provide install RHEL 5 with these options:

1. Install the OS in the English language (hwcert only works with English)
2. Choose a US keyboard.
3. Use the install number you provided on the "key" line. The install number unlocks the virtualization channels at install time, and these are necessary for certification.
4. Install a new instance of Red Hat Enterprise Linux Server (do not upgrade an existing installation)
5. Create a custom partition layout that meets the requirements of certification:
 - a. Delete any existing partitions
 - b. Create a 200MB **"/boot"** partition of type ext3. If you have multiple drives in the system, place this on the first drive
 - c. Create a 2000MB swap partition of type swap. This should also be on the first drive
 - d. Create a **"/"** (root) partition that uses the rest of the first drive. Choose the "Fill to maximum allowable size" option to use all the available space
 - e. Ensure that any other drives in the system are blank (no partitions)
6. Use the GRUB boot loader on the first drive (usually **/dev/sda**, this is the default setting).
7. Set all network devices to DHCP addresses, including those that do not come up at boot time. Note that wireless network cards have special requirements which are discussed in [Section A.1.23, "network"](#) of the test procedures chapter.
8. Use the time zone of your choice (you can change this to whatever you like in the kickstart file)
9. Use a root password of *redhat* by default (you can change this to whatever you like in the kickstart file)
10. Install the necessary packages to run hwcert (see the kickstart files for details)
11. Boot to the regular kernel (not kernel-debug) by default

We explain how to install the SUT using the kickstart file in [Section 2.18, "Configuring the System Under Test"](#).

⁸ http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Installation_Guide/index.html

⁹ http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Installation_Guide/ch-kickstart2.html

2.14. Preparing a Kickstart File to Install RHEL 6 and hwcert-client

The simplest way to install RHEL 6 and the hwcert packages is by using a kickstart file. We have prepared example kickstart files for a RHEL 6 hardware certification environment and made them available here on our FTP server:

- <ftp://partners.redhat.com/a166eabc5cf5df158922f9b06e5e7b21/hwcert/kickstarts/>

The files of interest are **rhel6-x86_64-client.ks** (RHEL 6, x86_64 architecture) and **rhel6-i386-client.ks** (RHEL 6, i386 architecture), which install a RHEL 6 system that is ready to undergo certification. These files can be duplicated, renamed, and edited to enable you to install different update releases of RHEL 6 (rhel6.1-x86_64-client.ks, rhel6.2-x86_64-client.ks, rhel6.3-x86_64-client.ks, and so on) in your test environment.

You can learn more about kickstart files in Chapter 32 of the [Red Hat Enterprise Linux 6 Installation Guide](#)¹⁰ entitled [Kickstart Installations](#)¹¹. To use a kickstart file to install your RHEL 6 SUTs and the test server, follow the directions in the Installation Guide referenced above to prepare a kickstart server, customize the example files to match your environment and copy them to your server. The kickstart files we provide install RHEL 6 with these options:

1. Install the OS in the English language (hwcert only works with English)
2. Choose a US keyboard
3. Install a new instance of Red Hat Enterprise Linux Server (don't upgrade an existing installation)
4. Create a custom partition layout that meets the requirements of certification:
 - a. Delete any existing partitions
 - b. Create a 500MB **"/boot"** partition of type ext4. If you have multiple drives in the system, place this on the first drive
 - c. Create a 2000MB swap partition of type swap. This should also be on the first drive
 - d. Create a **"/"** (root) partition of type ext4 that uses the rest of the first drive. Choose the "Fill to maximum allowable size" option to use all the available space
 - e. Ensure that any other drives in the system are blank (no partitions)
5. Use the GRUB boot loader on the first drive (usually **/dev/sda**, this is the default setting)
6. Set all network devices to DHCP addresses, including those that don't come up at boot time.
7. Use the time zone of your choice (you can change this to whatever you like in the kickstart file)
8. Use a root password of *redhat* by default (you can change this to whatever you like in the kickstart file)
9. Install the necessary packages to run hwcert (see the kickstart files for details)
10. Boot to the regular kernel (not kernel-debug or kernel-rt) by default

¹⁰ http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/index.html

¹¹ http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/ch-kickstart2.html

11. In RHEL 6 x86_64, install the MRG Realtime packages if desired (optional step)

We explain how to install the SUT using the kickstart file in [Section 2.18, “Configuring the System Under Test”](#).

2.15. Preparing a Kickstart File to Install RHEL 6 and hwcert-server

The simplest way to install a RHEL 6-based network and kdump test server is by using a kickstart file. We have prepared an example kickstart file and made it available here on our FTP server:

- [ftp://partners.redhat.com/a166eabc5cf5df158922f9b06e5e7b21/hwcert/kickstarts/](http://partners.redhat.com/a166eabc5cf5df158922f9b06e5e7b21/hwcert/kickstarts/)

The file of interest is **rhe16-server.ks**, which creates a RHEL 6 system that is ready to function as the network and kdump test server.

You can learn more about kickstart files in Chapter 32 of the [Red Hat Enterprise Linux 6 Installation Guide](#)¹² entitled [Kickstart Installations](#)¹³. To use this file to install your RHEL 6 test server, follow the directions in the Installation Guide referenced above to prepare a kickstart server, customize the example file to match your environment and copy it to your server. The kickstart file we provide installs RHEL 6 with these options:

1. Install the OS in the English language (hwcert only works with English)
2. Choose a US keyboard
3. Install a new instance of Red Hat Enterprise Linux Server (don't upgrade an existing installation)
4. Create a custom partition layout that meets the requirements of certification:
 - a. Delete any existing partitions
 - b. Create a 500MB **"/boot"** partition of type ext4. If you have multiple drives in the system, place this on the first drive
 - c. Create a 2000MB swap partition of type swap. This should also be on the first drive
 - d. Create a **"/"** (root) partition of type ext4 that uses the rest of the first drive. Choose the "Fill to maximum allowable size" option to use all the available space
 - e. Ensure that any other drives in the system are blank (no partitions)
5. Use the GRUB boot loader on the first drive (usually **/dev/sda**, this is the default setting)
6. Use the time zone of your choice (you can change this to whatever you like in the kickstart file)
7. Use a root password of *redhat* by default (you can change this to whatever you like in the kickstart file)
8. Install the necessary packages to function as a hwcert test server (see the kickstart files for details)
9. Disable selinux

¹² http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/index.html

¹³ http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/ch-kickstart2.html

10. Configure the server command to load a boot time by adding the command **hwcert-backend server start** to **/etc/rc.d/rc.local**

We explain how to install the test server using the kickstart file in [Section 2.17, “Configuring the network and kdump Test Server”](#).

2.16. UPDATED - Using Locally-hosted Pre-built Guest Files

The pre-built full-virt guest images contain everything you need in order to run the `fv_*` tests, either on RHEL 5 or RHEL 6, so you do not have to manually install a guest as part of the test procedure. The preconfigured image contains a RHEL 5 guest with **hwcert-client** installed as well as the logic necessary to pass test results from the guest out to the base system so that all test results can be delivered in a single results package.



Note

The pre-built guest image is RHEL 5-based, but it is used on both RHEL 5 Xen and RHEL 6 KVM. Separate images are not used for RHEL 5 and RHEL 6 virtualization testing.

The hwcert test suite will look locally on the SUT for the guest images whenever an `fv_*` test is launched. If they can't be found there, hwcert will attempt to download them from the network and kdump test server. If they can't be found there either, it will retrieve the images from the `partners.redhat.com` FTP site. Storing guest files on your network and kdump test server is useful if you are on a slow connection, or if you have to run the `fv_*` tests on more than one machine. Local retrieval is much faster than pulling several hundred megabytes worth of data from the Red Hat FTP servers every time you need to certify a system. The guest uses two files, which will need to be downloaded if you wish to serve the guest files locally. The files are:

1. The Xen guest image file
2. A data file that holds test results from the guest

There used to be a third file that was needed, a virt guest configuration file, but the tools now create that file for you when the image is imported.

We store the root URL for hwcert related files in **/usr/share/hwcert/lib/hwcert/environment.py**. The variable is **self.defaultPartnerServerURL**. The FV guest files are in the "RHEL5/FV-images" directory at that URL:

<http://partners.redhat.com/a166eabc5cf5df158922f9b06e5e7b21/hwcert/RHEL5/fv-images/>

This is how you use the above information to configure your network and kdump test server to host the FV guest images:

1. Set up a test server as explained in [Section 2.17, “Configuring the network and kdump Test Server”](#).
2. Create a directory on the server called **/var/www/hwcert/fv-images**
3. Copy the latest version of each of these files from the FTP site above into the local directory. The creation date of the files are part of their filename, so make sure to download the newest ones if multiples exist (the dates here are just examples):

- a. **hwcertData-20130409.img.tar.bz2** (domU->dom0 results transfer package, required for all arches)
- b. **v7i386-20120425.img.tar.bz2** (FV guest image for i386)
- c. **v7x86_64-20120424.img.tar.bz2** (FV guest image for x86_64)

Set up symlinks to the files as shown below:

1. **# ln -s hwcertData-20130409.img.tar.bz2 hwcertData.img.tar.bz2** (all arches)
2. **# ln -s v7i386-20120425.img.tar.bz2 hwcert-i386.img.tar.bz2** (i386)
3. **# ln -s v7x86_64-20120424.img.tar.bz2 hwcert-x86_64.img.tar.bz2** (x86_64)

As long as you specify a test server when prompted by **hwcert** before running any **fv_*** tests, your SUT will automatically retrieve the FV guest files from your local network and **kdump** test server instead of Red Hat's FTP server.



Note

The actual filenames for the guest image files still begin with "v7" for legacy purposes. Be sure to use "hwcert" in the names for your symlinks as shown above so they will work properly. We will be updating the real filenames from "v7" to "hwcert" at a later date.

2.17. Configuring the network and **kdump** Test Server

We will need a **network** and **kdump** test server before we can begin testing the SUT, so we can use that as our first installation. The installation procedure differs between the network test server and the SUT by only one item: the server needs to run the server tool after installation.



Important

The **hwcert-server** package will run on RHEL 6 only. This is a change from the earlier **v7-server** package that could run on both RHEL 5 and RHEL 6.

The system that functions as a **network** and **kdump** test server should be dedicated to the task. The machine does not need especially fast processors or large amounts of RAM, but its network connections must be at least as fast as those on the SUT. The test server and its NICs should be PCI Express-based if you are testing gigabit Ethernet or faster. It is recommended to locate the system as few hops away from the test systems on your network as possible; connecting it to the same network switch as the test systems is ideal. The server's processor architecture does not need to match that of the SUT, so you can use any available machine that meets the network speed criteria.

**Note**

The local test server can also store test results from the SUTs in preparation for upload to the Red Hat Hardware Catalog. This may be useful if your test lab has firewalled the SUT network and they cannot reach the Internet. If the network can be set up so that the local test server can reach the Internet and the SUTs can reach the local test server, results files can be copied from the SUTs to the local test server and then the results packages manually uploaded from there to the hardware catalog. Please see the section entitled [Section 3.7, “Submitting Results”](#) for more information on submitting results from the local test server.

To install the system, boot RHEL 6 using your preferred installation media (PXE/DVD/CD/USB key, etc.) and point the system at the kickstart file you set up in the previous section with this command at the "boot:" prompt:

```
boot: linux ksdevice=link ks=http://myserver.mydomain.com/kickstart_files/rhel6-server.ks
```

Replace the example shown here with your own path and kickstart file for the RHEL 6-based network and kdump test server.

The installation will complete unattended and you will be left with a fully configured RHEL 6 system that is ready to be used as a test server.

If you opted to not use the kickstart file and decided to install files manually, two steps must be performed to turn the system you just installed into a network test server. These commands must be run each time the machine is used as a server:

1. Log in and disable SELinux with the command **setenforce 0**.

To make this setting persistent across reboots, edit the file **/etc/selinux/config** and change the line **SELINUX=enforcing** to **SELINUX=disabled**.

2. Activate the test server with the command **hwcert-backend server start**

To make this setting persistent across reboots, edit the file **/etc/rc.d/rc.local** and add the command **hwcert-backend server start** to the end of the file.

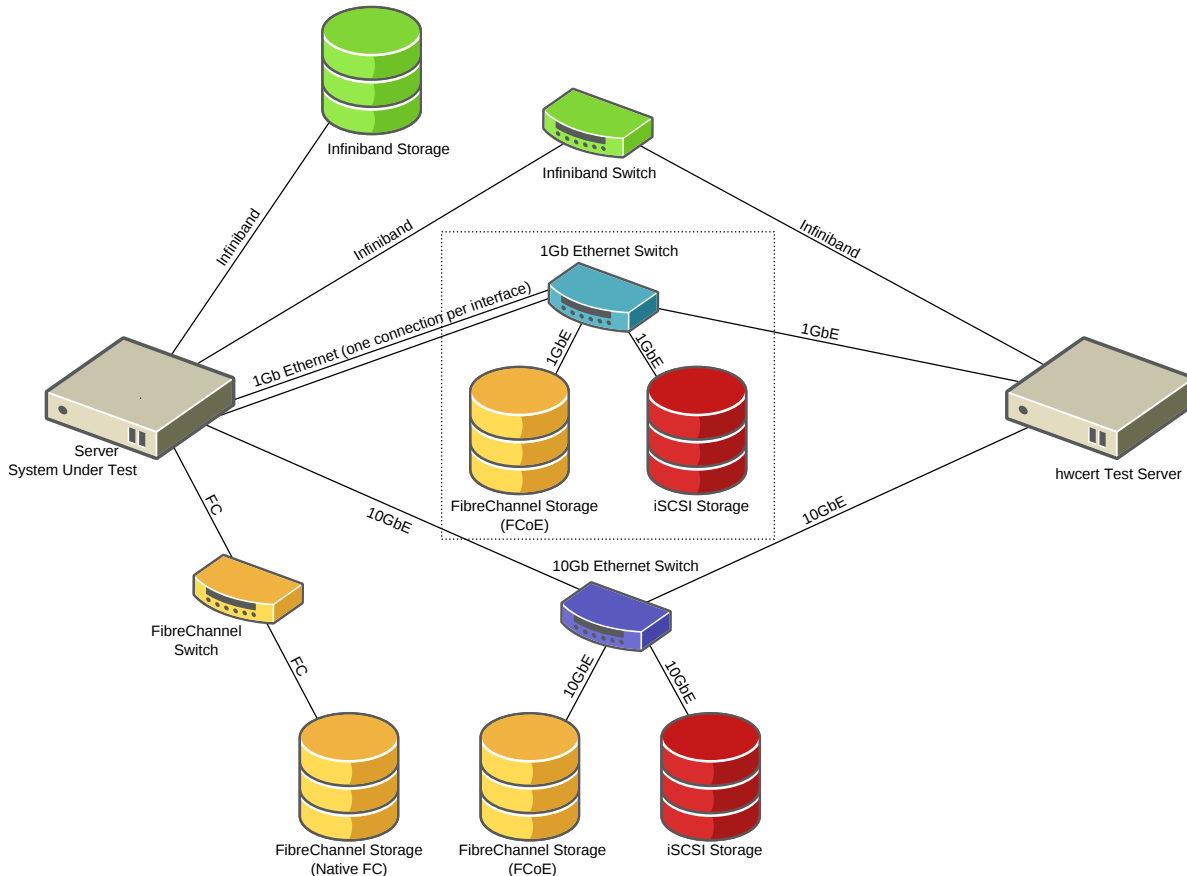
The server is now ready to perform the **network** and **kdump** tests. We will use it in the next sections after we have configured the SUT with appropriate hardware.

2.18. Configuring the System Under Test

Everything listed in the official test plan must be tested. Depending on the options available, this may require you to install and remove various pieces of hardware to run the same test multiple times. A system with multiple video card options is an example of this scenario, where multiple devices of the same class must be tested one at a time. In other cases you may be able to install multiple devices of the same kind at the same time and run the tests in the same test run. An example of this would be a system with enough drive bays and slots to hold two different SATA controllers and their associated drives. The order in which you install and test the components listed in the official test plan is up to you, we just need to receive passing results on all tests before we can post the system as certified.

We have created network and storage diagrams to help you connect your SUT to the hwcert test server and any required storage devices:

Connecting a Server



This example server offers two identical 1Gb Ethernet LOM interfaces (LAN-on-motherboard) and one 10Gb Ethernet network card that is classified as optional. All three of these cards offer FibreChannel over Ethernet (FCoE) and iSCSI support via hardware drivers. It has an optional Infiniband card that is listed in the specifications document as supporting both storage and network modes, and it also has an optional FibreChannel card that is advertised for use with external storage.

Note that none of the switches shown in this diagram are mandatory. Interfaces may be directly connected to the appropriate storage device or network device as required, but we recommend using switches where convenient as it simplifies device connections.

The Ethernet functionality of both 1Gb NICs needs to be tested as they are classified as integrated hardware. The 10Gb NIC also needs to be tested as it is classified as optional and is the only device offered that provides 10Gb connectivity.

FibreChannel over Ethernet (FCoE) storage testing is required on all three Ethernet NICs as they all support hardware-based FCoE. If the system's FCoE support was software-only, no testing would be required. Note that the same FCoE storage device can be used for both the 1GbE and 10GbE testing, provided it can connect at both speeds. The same storage device can also be used for FCoE and native FC testing if it supports both connection types.

Both storage and network testing are required for the Infiniband interface on this system, because the specifications document claims support for both storage and network connections.

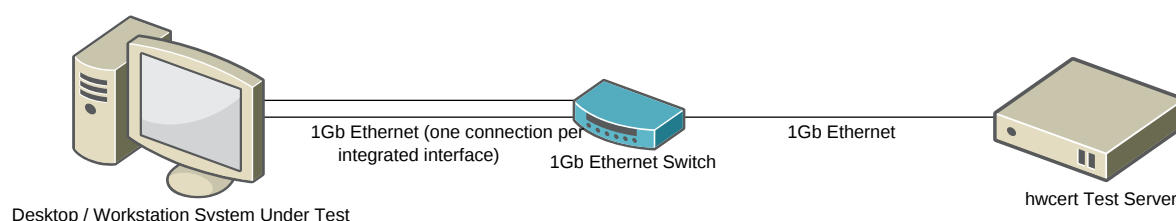
Native FibreChannel (FC) storage testing is required as the system's specifications document claims support for external FC storage devices. If the FC HBA were intended for use with internal storage, that would be tested instead. If the HBA could connect to internal and external storage, only one would need to be tested. The same storage device can be used for FCoE and native FC testing if it supports both connection types.

iSCSI storage testing is required on all three Ethernet NICs as they all support hardware-based iSCSI. If the system's iSCSI support was software-only, no testing would be required. The same iSCSI storage device can be used for both the 1GbE and 10GbE testing, provided it can connect at both speeds.

10Gb Ethernet was much more expensive than 1GbE at the time this diagram was created, so we have shown the use of a dedicated 10Gb Ethernet switch in this example. You could eliminate the 1Gb Ethernet devices inside the dotted-line box and connect all Ethernet devices through the 10Gb switch if:

1. You have sufficient ports on the 10GbE switch to handle all Ethernet connections
2. The 10GbE switch is capable of connecting at both 1Gb and 10Gb speeds

Connecting a Workstation

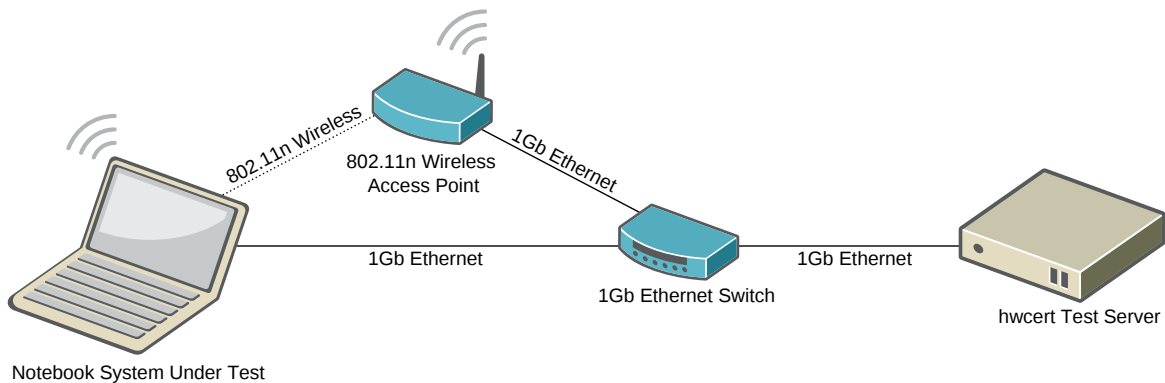


This example workstation has two identical 1Gb Ethernet LOM interfaces (LAN-on-motherboard) and no optional network devices. Both 1GbE cards must be connected and tested as they are classified as integrated hardware.

There can be multiple network devices between the SUT and the test server, so long as the speed of each connection is equal to or greater than the native speed of the interface(s) on the SUT. This rule applies when testing any kind of network device.

Best results will be obtained with the fewest number of devices between the systems, as this allows you to better control the amount of non-test traffic on the network. Too much extra traffic can lead to slower than expected test results, ultimately leading to a failed test. We recommend connecting the test server and the SUT to the same switch whenever possible for this reason. Direct connections between wired devices are also allowed, but using a switch will generally save you time and effort.

Connecting a Laptop



This example laptop system has 802.11n wireless Ethernet and a 1Gb Ethernet LOM interface (LAN-on-motherboard). There are no other available network options for this system. Both devices must be connected and tested as they are classified as integrated hardware.

A wireless router can take the place of a separate wireless access point and Ethernet switch for testing wireless devices. You may also use the wired Ethernet switch component of a wireless router when testing wired Ethernet devices. When selecting your network equipment, remember that the speed of all wired and wireless network connections throughout the test environment must be greater than or equal to the native speeds of the SUT.

A switch is not mandatory for wired network testing (direct connections can be made if you desire) but an intermediate network device like a wireless router or wireless access point is required for wireless testing as we do not support testing in Ad-Hoc mode.

Once you are comfortable with the hardware configuration of your system under test (SUT), install the system by booting RHEL using your preferred installation media (PXE/DVD/CD/USB key, etc.) and point the system at the appropriate kickstart file with this command at the "boot:" prompt:

```
boot: linux ksdevice=link ks=http://myserver.mydomain.com/kickstart_files/rhel6.4-client.ks
```

Replace the example shown here with your own path and kickstart file for RHEL 5 or RHEL 6.

The installation will complete unattended and you will be left with a fully configured RHEL 6 system that is ready to be certified. We will explain the actual process of certification in the next section.

Testing Systems and Components

3.1. Scheduling and Running Tests with **hwcert**

The new **hwcert** command automates many of the steps that had to be performed manually with the older **v7** tool. It calls another command called **hwcert-backend** that does the actual work of planning, testing, and communicating with the Hardware Catalog.

The **hwcert** command is capable of uploading results to the internet and fetching data about the mapping of kernel versions to RHEL releases. If the SUT requires a proxy to reach the Internet, environment variables like the following should be set before running the command:

```
export ftp_proxy="http://proxy.example.org:3128"
```

3.1.1. A Few Words on screen

The **screen** program is a text terminal window manager that allows you to, among other things, reconnect to sessions that you have been disconnected from. We recommend its use when running the **hwcert** tests through an ssh connection, as the network test will drop any active ssh connections when it restarts the network. It also allows you to log out of an ssh connection and leave **hwcert** running. Screen is very simple to use. The commands of importance to us are:

- Run **screen** to start the program. It will launch and provide a blank terminal in which you can perform your tests.
- Enter **ctrl-a**, **ctrl-d** to detach a screen session and return to the terminal. Any programs running inside the screen session will continue to run, even if you log out of the machine.
- Run **screen -r** to reconnect to a disconnected screen session.
- Type **exit** at the prompt while in a screen session to terminate screen.

3.1.2. Automated Testing of a New System

The first step that occurs on a newly installed SUT is the creation of the local test plan. This is not the same thing as the official test plan, the local test plan is the list of tests that **hwcert** can run based on all the hardware it detects in the SUT. You begin the process of creating the local test plan by running the command **hwcert**:

```
[root@localhost ~]# hwcert
The Red Hat Hardware Certification Test Suite
Hardware: Mycompany Ultraserwer X16
OS: Santiago 6.4
```

When prompted, you should confirm the proper vendor, make, URL, and category information for the system. In this example, we are certifying a server:

```
Please verify the hardware product information:
vendor:Mycompany
make:Ultraserwer
model:X16
product-url:http://www.mycompany.com/specs/ultraservers/X16
category (Desktop/Workstation|Laptop|Component|Peripheral|Server) Server
```

The next question determines how `hwcert` should use its automatic results upload feature. Choose one of these three options:

1. If you are certifying a new SUT, you have *not* already created a hardware certification request for it through the web interface or the `hwcert` tool, and you would like to upload the results to the Red Hat Hardware Catalog directly from the test tool itself, choose *new* and enter your hardware catalog (<http://hardware.redhat.com>) credentials when prompted.

```
What certification is this system being tested for? (new|existing|none) new
response: new
Red Hat Catalog User Name: myname@mycompany.com
response: myname@mycompany.com
Password:
Created certification ID 123456
http://hardware.redhat.com/show.cgi?id=123456
...
```



Note

Each certification entry can hold many results packages. You should only create new entries you are certifying a new model or component, or certifying an existing model or component on a different release of RHEL. You do not need to create separate entries each time you generate a results file or when you are testing different architectures of the same release of RHEL. For example, results from seven runs of the test suite on RHEL 6.3 x86_64 and i386 can be submitted to the same certification request for the Mycompany Ultraserver X16 server, but results from RHEL 5.8 x86_64 and RHEL 6.2 x86_64 would need to be submitted to separate requests.



Important

If you did not provide a specs URL when prompted, you must visit the hardware catalog entry for this system after it has been created to upload the specs document. If you do not, the review team cannot create an official test plan and cannot evaluate the results you submit. A link to the catalog entry is displayed for your convenience after the creation of the new certification request.

2. If you have already created a certification request for this system through the hardware catalog web interface or with the **`hwcert`** command and you would like to upload the results to the Red Hat Hardware Catalog directly from the test tool itself, use the *existing* option when prompted and enter the ID of the existing certification:

```
What certification is this system being tested for? (new|existing|none) existing
response: existing
Please enter the certification ID:123456
response: 123456
...
```

3. If you prefer to manually upload the results file through the hardware catalog web interface, your system cannot reach the hardware catalog via the Internet, or if this is just a test run that will not be submitted, enter *none* at the prompt:

```
What certification is this system being tested for? (new|existing|none) none
response: none
...
```

Now that the results submission method is settled, you will be prompted to enter the IP address or DNS name of the local test server, the system that runs the **hwcert-backend server** program and provides required communication for kdump and network testing:

```
Local Hardware Certification Test Server: mytestserver.mycompanylab.mycompany.com
```

If you chose *new* or *existing* as the type of certification, you will be asked if the system should check the catalog for the status of the tests in the official test plan. Answer *y* here to modify the local test plan based on the status of the official test plan:

```
Would you like to check for open tests for this certification? (y|n) y
```

You will be asked for your hardware catalog login information if you did not provide it earlier. If there is information on completed or required tests in the official test plan, hwcert will use that to adjust the local test plan and display only the tests that are needed to complete the certification.



Note

An error at this step indicating that the testplan is not frozen indicates that the official test plan has not been created for this request. This error can be safely ignored, but it means that no changes to the local test plan will occur as there is no official test plan to analyze.

This completes the initial information gathering. The system will now display the tests that hwcert has scheduled. We will be looking at full test plans in the next three sections, so here is an excerpt:

```
Created a new plan with 22 tests on 709 devices

The following tests are recommended to complete the certification:
1 -> USB2                pci0000:00/0000:00:01.1/0000:02:00.0/usb4
2 -> dvd                 sr0          host1/target1:0:0/1:0:0:0/block/sr0
3 -> cdrom               sr0          host1/target1:0:0/1:0:0:0/block/sr0
4 -> bluray              sr0          host1/target1:0:0/1:0:0:0/block/sr0
...
Ready to begin testing? (run|edit|quit)
```

The items in the plan are divided into columns:

- The first column is the number of the test.
- The second column shows whether the test is scheduled to be run ("->"), recommended to be skipped ("pp") due to previous passing results on the local system, not scheduled to be run (" ")

or always scheduled to be run ("`<>`"). If a test was not included or was marked as complete in the official test plan when you answered `y` to the question **Would you like to check for open tests for this certification? (y|n)**, the item would not appear in the local test plan. In this example from a completely new certification, all the tests are scheduled to be run.

- The third column is the name of the test.
- The fourth column is the name of the device on which the test will be run. For example, on test number four, the bluray test, device `sr0` will be tested. Not all devices will have a name in this column.
- The fifth item is the udev name of the device that the test will run on. For the bluray test item, the udev name is `"host1/target1:0:0/1:0:0/block/sr0"`. Again, not all test entries will have a name here.

You may begin your testing by answering `run` at the prompt. This will run every scheduled test, prompting for input during interactive tests (kdump, optical disc tests, storage etc.). Please see [Section A.1, "Test Procedures"](#) to learn which tests are interactive and which tests are completely automated.

When a run completes, you will see a summary of the test results, like this:

```
Test Run Summary:
```

```
-----
```

```
memory      - pass
video        - pass
profiler     - pass
info         - pass
...
```

Then you will be prompted to upload the results to the hardware catalog. If you answered `new` or `existing` to the earlier question **What certification is this system being tested for? (new|existing|none)**, you should enter `y` here to upload the results:

```
Would you like to submit the results to the hardware catalog? (y|n) y
compressing and encoding results...
uploading results...
done.
Created a new plan with 22 tests on 711 devices
Would you like to check for open tests for this certification? (y|n)
```

If you prefer to keep the results local, answer `n` when asked. You will be prompted to save the results to the local test server, which gives you a way to analyze tests results before uploading them to the hardware catalog (we will discuss results analysis in a later section):

```
Would you like to submit the results to the hardware catalog? (y|n) n
response: n
Saving current results to:
/var/hwcert/store/Mycompany/Ultraserer/X16/x86_64/RHEL/6.4/hwcert-results-
localhost.localdomain-20130806214636.xml.gz

Copy results to test server mytestserver.mycompanylab.mycompany.com? (y|n) y
sending file /var/hwcert/store/Mycompany/Ultraserer/X16/x86_64/RHEL/6.4/hwcert-results-
localhost.localdomain-20130806214636.xml.gz
response: Saved: /var/www/hwcert/store/Mycompany/Ultraserer/X16/x86_64/RHEL/6.4/hwcert-
results-localhost.localdomain-20130806214636.xml.gz
Upload took 0.22 seconds, 5.18 MB/sec
```



```
Created a new plan with 22 tests on 711 devices
...
```

The system will upload the results if you asked it to, create a new local test plan, check again for open tests on the hardware catalog if the results were uploaded, and return to the **Ready to begin testing? (run|edit|quit)** prompt. At this point, you can run the outstanding tests, edit the local test plan, which we will discuss in a moment in [Section 3.1.4, “Editing the Local Test Plan”](#), or quit. It is likely that the local test plan displayed now is the same as the local test plan you saw when you began your test run, as the reviewers are unlikely to have had time to review the results that were just submitted to satisfy test items. However, if you have submitted results before, it is recommended to answer *y* here to remove any tests that might have been marked complete.

3.1.3. Continuing Testing of a Partially Complete System

There are times when the first run of tests on a system does not complete all the testing due to required component swaps, reboots to use different kernels, or issues that cause tests to fail. In these cases you will need to run **hwcert** again. There are two methods to use here, depending on whether you are using **hwcert** to submit your results or you are submitting them manually:

Submitting Results With **hwcert**

- If the locally stored system information (vendor, make, model, etc.) still exists, when you run **hwcert** again the system will schedule a run of all the tests that do not yet have passing results. Answer *y* to the question **Would you like to check for open tests for this certification? (y|n)** to modify the local test plan to omit testing of any items that are not present in the official test plan or are marked as completed in the official test plan. Answer *run* at the prompt to begin your next round of testing, or *edit* to edit the local test plan. See [Section 3.1.4, “Editing the Local Test Plan”](#) for more information on editing the local test plan.
- If you performed a **hwcert clean all**, reinstalled, or otherwise lost your locally stored system information, running **hwcert** will begin the initial information gathering process again. When you get to the question **What certification is this system being tested for**, enter *existing* and the appropriate certification ID to link the system to the existing certification request:

```
What certification is this system being tested for? (new|existing|none) existing
response: existing
Please enter the certification ID:123456
response: 123456
...
```

Answer *y* to the question **Would you like to check for open tests for this certification? (y|n)** to modify the local test plan to omit testing of any items that are not present in the official test plan or are marked as completed in the official test plan. Answer *run* at the prompt to begin your next round of testing, or *edit* to edit the local test plan. See [Section 3.1.4, “Editing the Local Test Plan”](#) for more information on editing the local test plan.

Manually Submitting Results

- If the locally stored system information still exists, when you run **hwcert** again the system will schedule a run of all the tests that do not yet have passing results. If your SUT is capable of reaching the hardware catalog via the Internet, answer *y* to the question **Would you like to check for open tests for this certification? (y|n)** to modify the local test plan to omit testing of any items that are not present in the official test plan or are marked as completed in the official test plan. Answer *run* at the prompt to begin your next round of testing, or *edit* to edit the local test plan. See [Section 3.1.4, “Editing the Local Test Plan”](#) for more information on editing the local test plan.

- If you performed a **hwcert clean all**, reinstalled, or otherwise lost your locally stored system information, running **hwcert** will begin the initial information gathering process again. When you get to the question **What certification is this system being tested for**, enter *none*.

```
What certification is this system being tested for? (new|existing|none) none
...
```

Answer *run* at the prompt to begin your next round of testing, or *edit* to edit the local test plan. See [Section 3.1.4, “Editing the Local Test Plan”](#) for more information on editing the local test plan.

3.1.4. Editing the Local Test Plan

If you want to choose specific items in the local test plan to run, use the *edit* option at the **Ready to begin testing? (run|edit|quit)** prompt after running **hwcert**. When you answer *edit*, you will be taken to a screen that allows you to edit the items in the local test plan:

```
Ready to begin testing? (run|edit|quit) edit

1 -> USB2                pci0000:00/0000:00:01.1/0000:02:00.0/usb4
2 -> dvd                 sr0      host1/target1:0:0/1:0:0:0/block/sr0
3 -> cdrom               sr0      host1/target1:0:0/1:0:0:0/block/sr0
4 -> bluray              sr0      host1/target1:0:0/1:0:0:0/block/sr0
5 -> audio                pci0000:00/0000:00:02.0/0000:03:00.1/sound/card1
6 -> audio                pci0000:00/0000:00:1b.0/sound/card0
7 -> 1GigEthernet        eth1     pci0000:00/0000:00:1c.5/0000:09:00.0/net/eth1
8 -> 1GigEthernet        eth0     pci0000:00/0000:00:19.0/net/eth0
9 -> memory
10 <> info
Test Selection (<number>|<name(s)>|all|none|quit|run):
```

If you enter a test number or test name, the status of that test will switch between scheduled and non-scheduled. For our example system here, entering *1* would toggle the USB test from scheduled to non-scheduled:

```
1 -> USB2                pci0000:00/0000:00:01.1/0000:02:00.0/usb4
2 -> dvd                 sr0      host1/target1:0:0/1:0:0:0/block/sr0
3 -> cdrom               sr0      host1/target1:0:0/1:0:0:0/block/sr0
4 -> bluray              sr0      host1/target1:0:0/1:0:0:0/block/sr0
5 -> audio                pci0000:00/0000:00:02.0/0000:03:00.1/sound/card1
6 -> audio                pci0000:00/0000:00:1b.0/sound/card0
7 -> 1GigEthernet        eth1     pci0000:00/0000:00:1c.5/0000:09:00.0/net/eth1
8 -> 1GigEthernet        eth0     pci0000:00/0000:00:19.0/net/eth0
9 -> memory
10 <> info
Test Selection (<number>|<name(s)>|all|none|quit|run):
```

The option *all* will schedule all available tests to be run, and the option *none* will remove all tests from the scheduled list (except for the info test, which always runs).

Enter *run* when you have finished editing the test plan and want to test the scheduled items, or enter *quit* to exit the test tool.

3.2. Certifying for Red Hat Enterprise Linux

In this section, we will describe the tests that are required to certify systems and components for use with Red Hat Enterprise Linux. They will be introduced in the context of certifying a server, a desktop/workstation, a laptop, and a component.

3.2.1. Testing a Server

Most of the hardware found on server systems can be found on other types of computers, but servers tend to have larger amounts of common components like disks, RAM, and CPUs. We can also see a tape drive, something that is generally found only on servers, in the **hwcert** output from this example server. We will begin walking through all the hardware certification tests, starting with the items that are most commonly associated with servers. We have marked these with "***" in the test plan below. You can find detailed information on all the tests in [Section A.1, “Test Procedures”](#) if you want to learn more. Remember, even though we are only covering a few tests in this example, you will need to submit results for everything listed in the official test plan, regardless of what type of computer you are testing.

```
The following tests are recommended to complete the certification:

1 -> USB2                pci0000:00/0000:00:01.1/0000:02:00.0/usb4
2 -> dvd                  sr0          host0/target0:0:0/0:0:0:0/block/sr0
3 -> cdrom                sr0          host0/target0:0:0/0:0:0:0/block/sr0
4 -> cdrom                sr1          host4/target4:0:1/4:0:1:0/block/sr1
5 -> 10GigEthernet       eth1         pci0000:00/0000:00:1c.5/0000:06:00.0/net/eth1
6 -> 1GigEthernet        eth0         pci0000:00/0000:00:1c.4/0000:05:00.0/net/eth0
** 7 -> memory
** 8 -> fv_core
** 9 -> fv_storage
** 10 -> fv_memory
** 11 -> fv_network
** 12 -> fencing
13 -> video               pci0000:00/0000:00:1c.0/0000:02:00.0/0000:03:02.0/00
14 -> storage             host4        host4/target4:0:0/4:0:0:0/block/sdb
15 -> storage             host2        host2/target2:0:0/2:0:0:0/block/sda
** 16 -> cpuscaling
** 17 -> core
** 18 -> profiler
** 19 <> info
** 20 -> kdump            nfs
** 21 -> kdump            local
** 22 -> tape             st0
Ready to begin testing? (run|edit|quit)
```

RAM

System RAM is tested with the **memory** test. It does not cover things like USB flash memory, SSD drives, proprietary flash caches or any other type of memory device, just the RAM as used by the kernel's virtual memory (VM) subsystem. The [Policy Guide](#)¹ explains how much memory should be used when performing the memory test.

¹ <https://hardware.redhat.com/doc/index.html>

Virtualized Guest Tests

On all virt-capable architectures of RHEL 5 and RHEL 6, test plans generated on virt-capable machines will include a number of tests beginning with "fv_":

- **fv_core** - Tests a pre-built RHEL 5 guest's vCPU
- **fv_memory** - Tests a pre-built RHEL 5 guest's memory
- **fv_network** - Tests a pre-built RHEL 5 guest's network connection (optional)
- **fv_storage** - Tests a pre-built RHEL 5 guest's storage device (optional)

These are special tests that are designed to run on fully-virtualized (FV) Xen (RHEL 5) and KVM (RHEL 6) guests. The actual test portion of the scripts is identical to that of our regular **core**, **memory**, **network** and **storage** tests, but the fv_ scripts have additional code in them that downloads a pre-built FV guest from Red Hat (if it is not already present on the SUT or your network test server), launches it, and then runs the tests in the guest. Please see the appropriate section in the server explanation (this section) and [Section 3.2.2, "Testing a Desktop/Workstation"](#) for information on the tests that are run by these scripts.

CPU

The CPU is obviously used in all testing, but there are a few tests in **hwcert** dedicated to analyzing specific attributes of the CPU itself. These are the **core**, **cpuscaling** and **profiler** tests. We will go through them in that order.

The **core** test examines the system's CPUs and ensures that they are capable of functioning properly under heavy loads. The **cpuscaling** test exercises the CPUs at varying frequencies to make sure that they can increase and decrease clock speed as required by the compute load. The **profiler** test examines the CPUs' ability to run the **oprofile** monitoring tool. It is a low-overhead performance monitoring tool that is used to uncover performance related issues and is extensively used by large shops with large RHEL deployments. The [Policy Guide](#)² tells you how to choose the proper CPUs for testing, in the event that the system ships with multiple CPU options.

fencing

The **fencing** test is designed to test the ability of a system to be powered cycled remotely, either by the system's baseboard management controller (BMC) or an external fencing device like an APC or WTI power device. Fencing is a process that most commonly occurs when a system is a member of a cluster. If the system is determined by the other members of the cluster to be operating incorrectly it can be fenced, or forcibly rebooted, to attempt to restore it to a normal working state.

Info

The **info** test is different from the other tests in the **hwcert** test suite. It is not testing a specific component or testing a particular system function, its job is to gather information about the operating system and data on all the installed hardware any time a test is executed. We use the data in the info test to confirm that only approved kernel modules are being used and that the hardware being tested matches the hardware listed in the official test plan. It is also useful to us when helping to debug failed tests. **info** automatically runs any time **hwcert** tests are executed and it is a required part of all results packages.

² <https://hardware.redhat.com/doc/index.html>

kdump (RHEL 6 only)

The **kdump** test performs two tests to ensure that the system can capture a vmcore when a system crash occurs. The vmcore file generated by kdump is an important part of the crash debugging process performed by our support team, so it is important to ensure that a system can create one in the event of a crash. It will save the file locally and also to an NFS server (the **kdump/network** server) to test the most common methods of core file access.

Tape Drive

The **tape** test covers testing of tape drive devices. It can test robotic tape libraries as well, but testing of the robot itself is outside its scope. The test is a simple rewind, write and compare.

3.2.2. Testing a Desktop/Workstation

Desktop and workstation systems have a few items, like audio cards, that are not typically found in servers, but the rest of the hardware is virtually identical. The most common differences are less RAM, fewer disk options, fewer processors and a more robust video card. Here is a local test plan from a workstation where we have tagged with "***" the tests we will be discussing in this section. Full details on all the test procedures can be found in [Section A.1, "Test Procedures"](#). Remember, even though we are only covering a few tests in this example, you will need to submit results for everything listed in the official test plan, regardless of what type of computer you are testing.

```

The following tests are recommended to complete the certification:

** 1 -> USB2                pci0000:00/0000:00:01.1/0000:02:00.0/usb3
** 2 -> USB3                pci0000:00/0000:00:01.1/0000:02:00.0/usb4
** 3 -> dvd                 sr0      host1/target1:0:0/1:0:0:0/block/sr0
** 4 -> cdrom               sr0      host1/target1:0:0/1:0:0:0/block/sr0
** 5 -> bluray              sr0      host1/target1:0:0/1:0:0:0/block/sr0
** 6 -> audio               pci0000:00/0000:00:02.0/0000:03:00.1/sound/card1
** 7 -> audio               pci0000:00/0000:00:01b.0/sound/card0
** 8 -> 1GigEthernet        eth1     pci0000:00/0000:00:1c.5/0000:09:00.0/net/eth1
** 9 -> 1GigEthernet        eth0     pci0000:00/0000:00:19.0/net/eth0
10 -> memory
11 -> fv_storage
12 -> fv_memory
13 -> fv_network
14 -> fv_core
** 15 -> video               pci0000:00/0000:00:02.0/0000:03:00.0
** 16 -> storage             host5    host5/target5:0:0/5:0:0:0/block/sdb
** 17 -> storage             host0    host0/target0:0:0/0:0:0:0/block/sda
18 -> profiler
19 -> cpuscaling
20 -> core
21 <> info
22 -> kdump                 nfs
23 -> kdump                 local
Ready to begin testing? (run|edit|quit)

```

USB Ports

Virtually every modern computer contains USB2 and/or USB3 ports. We use a simple plug/unplug action in the **USB2** and **USB3** tests to make sure that the system recognizes when hardware is plugged into and removed from every available port on the system. We know that you are probably using one or two USB ports for a keyboard/mouse/KVM during the test, so we allow you to specify the number of ports that are available at the start of the test. Just make sure to subtract any that are in use

when you are asked for the number. You will need to test each port with the appropriate kind of USB device (USB2 devices for USB2 ports, and USB3 devices for USB3 ports).

Blu-Ray, DVD and CD Drives

The **bluray**, **dvd** and **cd** tests test the rewrite, write and read capabilities of the three different types of optical media. The rules of hardware certification require only one of these tests per set of drives that is offered with a controller. You now run a test on any drive that has the most advanced feature set of all the drives available for each controller. See [Section A.1.6, “bluray”](#) for more information on how to choose the proper drive and the proper test.

Sound Cards

The **audio** test covers both integrated and removable sound cards and tests both with a simultaneous playback and record. For such a simple test, it can take a fair amount of configuration testing to achieve a passing result. We recommend reading the detailed directions in [Section A.1.4, “audio”](#) to learn how to set everything up for a successful test.

Network Cards

The **1GigEthernet** test tests gigabit Ethernet devices it detects with a variety of throughput and ping tests. You need to have the network server running before you start the test and specify it by IP or DNS name when prompted. In addition to the test for gigabit Ethernet cards, there are also tests for 10 gigabit Ethernet (**10GigEthernet**) and 100 megabit Ethernet (**100MegEthernet**) that will be planned automatically if those devices are detected. If you are testing a wireless network card, see [Section A.1.23, “network”](#) for configuration requirements for those cards.

Video Cards

The **video** test covers all video hardware, integrated or add-in, and involves hardware detection, X11 configuration file creation and a basic X11 server performance test. We recommend that you connect your monitor directly to the system (avoid using a KVM) if you encounter trouble with monitor detection during the test.

IDE, SATA, SAS, SCSI Drives, SSD Drives and Media Cards

The **storage** test performs a variety of read and write tests on all kinds of storage devices. The same test script is run regardless of drive type. The number of **storage** runs and the devices required to be tested during each run is dictated by our “test all code paths” requirement, which requires testing of all the different storage drivers that can be used by the storage options available with a system. We recommend reading [Section A.1.28, “storage”](#) for more information.

3.2.3. Testing a Laptop

Laptop systems have a few devices that generally are not found in other systems. These include built-in screens, PC Card sockets, ExpressCard sockets, batteries and wireless network cards. Here is the local test plan from an example laptop computer where we have tagged with “***” the tests that we will be discussing in this section. Full details on all the test procedures can be found in [Section A.1, “Test Procedures”](#). Remember, even though we are only covering a few tests in this example, you will need to submit results for everything listed in the official test plan, regardless of what type of computer you are testing.

The following tests are recommended to complete the certification:

```

** 1 <> suspend          device:09/PNP0C09:00/PNP0C0A:00/power_supply/BAT0
2 -> USB2              pci0000:00/0000:00:01.1/0000:02:00.0/usb3
** 3 -> battery         device:09/PNP0C09:00/PNP0C0A:00/power_supply/BAT0
** 4 -> lid             LNXSYSTM:00/LNXSYBUS:00/PNP0C0D:00/input/input0
5 -> cdrom             sr0      host1/target1:0:0/1:0:0:0/block/sr0
6 -> dvd              sr0      host1/target1:0:0/1:0:0:0/block/sr0
7 -> audio            pci0000:00/0000:00:1b.0/sound/card0
** 8 -> expresscard
** 9 -> pccard
10 -> Ethernet        pan0      virtual/net/pan0
** 11 -> WirelessN    wlan0     pci0000:00/0000:00:1c.4/0000:02:00.0/net/wlan0
12 -> 1GigEthernet    eth0      pci0000:00/0000:00:19.0/net/eth0
13 -> memory
14 -> fv_memory
15 -> fv_core
16 -> fv_network
17 -> fv_storage
18 -> video           pci0000:00/0000:00:02.0
19 -> storage         host0     host0/target0:0:0/0:0:0:0/block/sda
20 -> core
21 -> cpuscaling
22 -> profiler
23 <> info
24 -> kdump          nfs
25 -> kdump          local
Ready to begin testing? (run|edit|quit)

```

Built-in Battery

Two of the hwcert tests are targeted at systems with a built-in power source. They are the **suspend** test and the **battery** test. The **suspend** test tests entering and leaving S3 suspend (suspend to RAM) and S4 hibernate (suspend to disk) states. It enters these states with the **pm-suspend** command and then prompts you to manually trigger sleep and hibernate using the keyboard. These tests cannot run with the Xen kernel on RHEL 5, as it does not support suspend or hibernate. You will need to reboot into the bare metal kernel to run the **suspend** test if you are certifying the system on RHEL 5. Note: This test will be scheduled once per run on laptop systems. This is to ensure that all hardware is available and functional after resuming.

Battery tests whether the integrated battery is usable as the system's primary power source and also confirms that it can be charged via the AC adapter. The test is only run for systems with built-in batteries that are designed to power the entire system. It is not designed to test UPS-style backup batteries or batteries on RAID controllers. You will be prompted to unplug and plug in the AC-adapter during the test.

Built-in Screen

The **lid** test is used on systems with an integrated screen to ensure that the backlight automatically turns off whenever the lid (screen) is closed. It is a manual test that relies on the tester to confirm that the screen's backlight really turns off when the lid is closed.

ExpressCard Slots

ExpressCards are a novel form factor that combines USB and PCI Express interfaces. The **expresscard** test ensures that any ExpressCard slots in the system can communicate through both interfaces. You will be prompted to insert and remove the card(s) during the test.

PC Card Slots

PC Cards (originally known as PCMCIA cards) are an older-style card interface that is generally only found on laptops. The **pccard** test ensures that we can count any PC Card sockets that are present in the system, that we can turn them on and off and that we can probe them for information about the card(s) they contain. You will need to insert a card in your socket(s) before starting any runs of **hwcert** that contain a **pccard** test.

Wireless Network Adapter

The **WirelessN** test tests any 802.11n wireless Ethernet devices with a variety of throughput and ping tests. Wireless devices are generally only found on laptops, but they are required to be tested wherever they are found. We also have a test called **WirelessG** that will be planned if any 802.11g wireless Ethernet devices are present. There is a section in [Section A.1.23, “network”](#) that explains exactly how to set up a wireless card to be tested with **WirelessN** or **WirelessG**.

3.2.4. Testing a Component

Component certifications are generally performed on the same type of system (server/desktop/laptop) for which the component is intended, like a server for a high-end SSD drive or a desktop for a value-line video card. The test system needs to fully support all the features of the component, so make sure that your test system has the necessary capacity to handle the component(s). The test plan below is similar to the plan from our example server system from a few sections back, but it we will use it to illustrate the testing of a tape device component listing. We are looking at the item we have tagged with “*” for this example. Full details on all the test procedures can be found in [Section A.1, “Test Procedures”](#). Remember, even though we are only covering a single component in this example, you will need to submit results for every test listed in the official test plan, regardless of what type of component you are testing.

The following tests are recommended to complete the certification:

```
1 -> USB2                pci0000:00/0000:00:01.1/0000:02:00.0/usb3
2 -> dvd                 sr0      host0/target0:0:0/0:0:0/block/sr0
3 -> cdrom               sr0      host0/target0:0:0/0:0:0/block/sr0
4 -> cdrom               sr1      host4/target4:0:1/4:0:1:0/block/sr1
5 -> 1GigEthernet        eth1     pci0000:00/0000:00:1c.5/0000:06:00.0/net/eth1
6 -> 1GigEthernet        eth0     pci0000:00/0000:00:1c.4/0000:05:00.0/net/eth0
7 -> memory
8 -> fv_core
9 -> fv_storage
10 -> fv_memory
11 -> fv_network
12 -> fencing
13 -> video              pci0000:00/0000:00:1c.0/0000:02:00.0/0000:03:02.0/00
14 -> storage            host4     host4/target4:0:0/4:0:0:0/block/sdb
15 -> storage            host2     host2/target2:0:0/2:0:0:0/block/sda
16 -> cpuscaling
17 -> core
18 -> profiler
19 <> info
20 -> kdump              nfs
21 -> kdump              local
** 22 -> tape            st0
Ready to begin testing? (run|edit|quit)
```


Tape Drive

The **tape** test covers testing of tape drive devices. It can test robotic tape libraries as well, but testing of the robot itself is outside its scope. The test is a simple rewind, write and compare. As this is a component test, you will need to use the *edit* option to select **tape** and deselect other test if the official test plan has not been created and your SUT is not checking the hardware catalog for the list of tests to run.

Performing Multiple Component Certifications

When you certify an entire system, you want the certification ID that you entered or that was created during the **hwcert** run to be persistent. Any additional runs you make on that system for failed test reruns or additional hardware coverage should go into the same certification entry. The only time you want the ID to change is when you certify for another major release of the operating system, as that needs to occur in a different certification request. The process of installing the other release will wipe out everything on the drive, including file where the old certification ID is stored.

When performing component certifications, you might use a single system to test two or more components without reinstalling the OS. A persistent certification ID would result in you uploading the results files for all your devices to the same certification entry if you use the automatic upload feature. This is not what you want to do if you want individual component certs.

Here are two options to work around this problem:

1. You can use the manual results upload process to generate a results file and then attach it to the correct certification request using the web interface. See [Section 3.7, “Submitting Results”](#) for more information.
2. You can use the command **hwcert clean all** to delete the old plan and results after uploading all your results files for the first component, then generate a new plan with **hwcert** for the second component. This will allow you to use the automatic upload process to submit all your results.

3.3. Certifying for Red Hat Enterprise Linux Open Stack Platform Compute

An option to create a certification entry for Red Hat Enterprise Linux OpenStack Platform (RHEL OSP) appears at the end of any system certification request that is opened for Red Hat Enterprise Linux 6, x86_64 architecture, up to and including version 6.4. You may opt out of the automatic RHEL OSP certification by selecting the *No* option, but you should know that no additional tests beyond those normally required for RHEL certification are needed to achieve RHEL OSP Compute certification. If you allow the RHEL OSP certification request to be created, your system will be marked certified for both RHEL and RHEL OSP on successful completion of the RHEL certification, assuming you pass the virtualization tests, which are a standard part of the RHEL cert. If you have any questions about this, please contact your Red Hat Support representative.

3.4. Certifying for Red Hat Enterprise MRG Realtime

MRG Realtime is part of the Red Hat Enterprise MRG (Messaging, Realtime, Grid) add-on to Red Hat Enterprise Linux. It is used for time-critical workloads that need to execute in a defined, predictable way. Any server that is certified to run Red Hat Enterprise Linux 6, x86_64 architecture, is eligible to be certified for MRG Realtime. If you are interested in certifying a system to run MRG Realtime, confirm that it is already certified for RHEL 6, x86_64 architecture, then talk to your Red Hat Support representative to learn more about MRG Realtime. Follow these directions when you are ready to begin MRG Realtime certification:

1. Follow the optional MRG Realtime section in [Section 2.12, “Preparing the Test Environment”](#) to get all the necessary MRG Realtime packages installed on your NFS or HTTP server.
2. Configure the hardware on your server, being sure to follow the System Processors and System Memory guidelines in section 4.7 of the Hardware Certification Policy Guide located here: https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Hardware_Certification/1/html-single/Policy_Guide/index.html.
3. Perform a fresh install of RHEL 6 x86_64 and the hwcert packages on your test system, including the Realtime kernel and its dependencies:
 - kernel-rt
 - kernel-rt-debuginfo
 - kernel-rt-debuginfo-common-x86_64
 - rt-setup
 - rtctl
 - rteval
 - rteval-loads
 - rt-tests
 - libxslt-python

If you use a kickstart file based on the rhel6-x86_64-client.ks RHEL 6, x86_64 kickstart file referenced in [Section 2.14, “Preparing a Kickstart File to Install RHEL 6 and hwcert-client”](#), uncomment and configure the MRG Realtime sections (the yum repos and the packages) to have these packages installed for you from your NFS or HTTP server.

4. Boot into the kernel-rt kernel.
5. Run **hwcert** to create the new test plan. Be sure to enter *existing* or *none* when prompted for the type of certification you are performing, as new MRG Realtime certification requests must be opened through the hardware catalog's web interface (<http://hardware.redhat.com>). Directions for opening a MRG Realtime certification are listed below. The hwcert program will recognize the realtime kernel and add two new tests, **hwlatdetect** and **rteval**.
6. Run the two necessary MRG Realtime tests (plus the always-required info test). The **hwlatdetect** test takes approximately two hours and the **rteval** test takes about twelve.

Note that you may need to enlist the help of your company's BIOS engineers to tune the system in order to meet the strict latency requirements of MRG Realtime. Any BIOS or other tuning parameters (disable hyperthreading, shut off hardware monitoring, etc.) required to pass MRG Realtime certification would need to be posted on a public web page that can be linked to by a Red Hat Knowledgebase article. That knowledgebase article would be associated with your MRG Realtime certification entry, ensuring that any customer wanting to run your certified MRG Realtime system would know exactly what tuning to perform in order to get the highest levels of performance on the Realtime kernel.

Follow these steps to create a request and submit results for a MRG Realtime certification:

1. Log in to the hardware catalog at <http://hardware.redhat.com>.

2. Search for the system that you wish to certify for MRG Realtime and click its listing to view the full certification information. If you already know the direct URL for the hardware, load that in your browser instead of searching. Note that the initial Red Hat Enterprise Linux certification request for the system must be complete before you can submit a request for MRG Realtime certification.
3. Click the **Advanced** link near the top of the certification entry.

4.

Create New Layered Product Certification - MRG

Vendor	Mycompany
Make	Ultraserver
Model	X16
Product	Red Hat Enterprise MRG
Version	MRG Realtime v2.0 ▾

CREATE MRG

Under **Create new Layered Production Certification**, choose the appropriate MRG Realtime version in the dropdown box and then click **Create MRG** to create the request.

5. You will automatically be taken to the new MRG Realtime certification request.

From this point, the MRG Realtime certification request proceeds like a Red Hat Enterprise Linux request, with the manual or automatic submission of results files, completion of the test plan items, discussion with the review team, etc.

3.5. Certifying for Red Hat Storage

Please contact your Red Hat Support representative if you are interested in certifying for Red Hat Storage. The main step in certifying a system for the Red Hat Storage product is the completion of a Red Hat Enterprise Linux 6 certification as described in [Section 3.2, “Certifying for Red Hat Enterprise Linux”](#). The other criteria for Red Hat Storage certification are described in a Red Hat Knowledgebase article that can be viewed at this link: <https://access.redhat.com/site/articles/66206>.

3.6. Troubleshooting Failed Tests

We will use this space to document workarounds for common test failures.

- Q: Why did the hwcert tool not detect any CPUs for the cpuscaling test? I know my processors have CPU frequency scaling capabilities.

A: First, make sure CPU frequency scaling is enabled in the BIOS. If it is turned on and the **hwcert** command does not schedule the test it is probably caused by a BIOS that does not advertise the scaling ability of the CPU(s). See [Section A.1.9, “cpuscaling”](#) in the test procedure guide for more information and for directions on manually adding the test.

- Q: The profiler test keeps failing. Is there anything I can try to get a passing result?

A: Some systems require a boot-time option in order for the test to complete. Please see [Section A.1.25, “profiler”](#) in the test procedure guide for more information and for directions on manually adding the test.

- Q: Why does the network test always fail with SELinux errors appearing on the network test server?

A: SELinux must be disabled on the network test server before the test is run on the SUT. Please see [Section 2.17, “Configuring the network and kdump Test Server”](#) for more information.

- Q: Why do the suspend tests fail under the Xen kernel?

A: The Xen kernel does not support suspend or hibernate actions, so the test will always fail if run under the Xen kernel. Please see [Section A.1.29, “suspend”](#) in the test procedure guide for information on running under the bare metal kernel.

3.7. Submitting Results

The **hwcert** program prepares results files for submission, either by automatic upload to the hardware catalog or by the creation of a results XML file that you can manually upload through the hardware catalog web interface. We have talked about both methods previously in the documentation, but we will review them here.



Important

If you are certifying a system for MRG Realtime, you will need to use the *existing* option and provide an existing MRG Realtime certification request or submit your test results manually. You cannot use the *new* option due to the way MRG Realtime certifications must be created. Please see [Section 3.4, “Certifying for Red Hat Enterprise MRG Realtime”](#) for more information.

3.7.1. Automatic Upload from the SUT

As mentioned earlier, results can be automatically uploaded to a certification entry on the hardware catalog by the **hwcert** application on the SUT. If you answered *new* or *existing* to the question **What certification is this system being tested for? (new|existing|none)** the system will prompt you to upload the test results at the end of each test run:

```
...
Your sosreport has been generated and saved in:
/tmp/sosreport-localhost-20130811170703-3230.tar.xz
Copied sosreport --batch -n selinux /tmp/sosreport-localhost-20130811170703-3230.tar.xz to /
var/log/hwcert/runs/1/info
PASS
copying attachments...
checking directory /var/log/hwcert/runs/1/info
Skipping output.log
Adding /var/log/hwcert/runs/1/info/sosreport-localhost-20130811170703-3230.tar.xz
copied attachment file sosreport-localhost-20130811170703-3230.tar.xz
saveOutput: /var/log/hwcert/runs/1/info/output.log
Return value was 0

Would you like to submit the results to the hardware catalog? (y|n)
```

Answering *y* will upload the data to the hardware catalog, clean the results and create a new local test plan.

```
response: y
compressing and encoding results...
uploading results...
done.
Created a new plan with 22 tests on 711 devices
Would you like to check for open tests for this certification? (y|n)
```

If you have more tests to perform, we recommend that you answer *y* to the "check for open tests" question and let the system create a local test plan with only the items that are remaining in the official test plan. Then, when you are returned to the **Ready to begin testing?** prompt, run the remaining tests. If you are finished with testing, you can answer *n* here and then *quit* when you are returned to the **Ready to begin testing?** prompt

You do not need to follow the directions in the following "Manual Submission" section if you just submitted your results file using the automatic process.

3.7.2. Manual Upload

If you specified *none* when asked what kind of certification you were creating, you will be given another chance to create a new cert, upload the results to an existing cert, or continue with the manual results file creation and upload process when your run of the `hwcert` command completes.

```
Would you like to submit the results to the hardware catalog? (y|n)
```

If you have changed your mind and would like to use the automatic upload procedure, answer *y* and then follow the on-screen prompts to create a new certification entry or link to an existing one, and then follow the directions in [Section 3.7.1, "Automatic Upload from the SUT"](#) to submit the results.

If you still wish to create a results package you can submit manually, answer *n* to the question and the `hwcert` application will create a results file and save it to the `/var/hwcert/store` directory:

```
response: n
Saving current results to:
/var/hwcert/store/Mycompany/Ultraservert/X16/x86_64/RHEL/6.4/hwcert-results-
localhost.localdomain-20130811214150.xml.gz
```

That `.xml.gz` file is the results file you upload to the hardware catalog. We will explain how to do that in just a moment, but first we need to deal with the next question that the `hwcert` application asks you:

```
Copy results to test server mytestserver.mycompanylab.mycompany.com? (y|n)
```

Storing a copy of the results file on the local test server is useful for two reasons:

1. You can analyze the test results on the local test server through a web browser just as you would on the Red Hat Hardware Catalog site (<http://hardware.redhat.com>). Directions for analyzing the results file in either location will be added later.
2. If you have set up your lab such that the local test server has Internet access, the server can act as a local store and forward location for results files from all your servers. You can leave the SUTs

behind the firewall and use the local test server to submit results files to the Red Hat Hardware Catalog. We are investigating the enhancement of the **hwcert** command to allow testers to push results to the catalog from the local test server, but for now it is a manual process using the hardware catalog's web interface. The results files will be in the server's **/var/hwcert/store** directory if you allowed the **hwcert** command to transfer the results to the test server.

After answering the question about copying the results to the test server, the system will upload the file to the local test server if you requested it to do so, then return to a fresh local test plan. Otherwise it will skip the upload and return to a fresh local test plan. If you would like to continue testing, edit the local test plan to choose the tests to run and then run them, or enter *quit* to exit.

Now you can upload the results file to the hardware catalog. Each certification request can have as few as one or as many as dozens of results packages attached to it, depending on the system's complexity. We want all certification results for a particular system that are run on the same version of RHEL attached to the same ticket. This means that 32-bit and 64-bit results packages go on the same request, as do re-runs to correct for failed or missing tests, but tests for different versions of RHEL would need to go in different certification requests. There is a size limitation of 16MB for each results file. If you have a results package greater than 16MB in size, please contact your Red Hat Support representative for further assistance.

You attach results packages by going to the certification entry on the hardware catalog, scrolling down to the upload section, and filling out the requested information:

UPLOAD RESULTS PACKAGE OR SPECIFICATION

File

<input type="text"/>	SELECT
----------------------	--------

Currently supported test suite versions: **hwcert-dient-1.6.4-57**, v7-1.6.4-22. Currently acceptable maximum result file size: **16MB**

Description

Describe the attachment briefly.

Comment

(optional) Add a comment about this attachment to the certification.

Attachment Type

☐ Results

☐ Specification Document

Upload Results Package or Specification Document Fields

- **File** - The location of the results package. If you are uploading from the SUT, the file will be in the `/var/hwcert/store` directory.

- **Description** - A brief description of the package. It is useful to put things here like the model of the SATA controller being tested if you are submitting tests from several different controllers on this system.
- **Comment** - This is a field for you to use when you need to provide more information than just the one line in the Description.
- **Attachment Type** - Tell us whether you are uploading a results package, a specifications document or some other kind of file.

When you have provided the requested information, click the **Select File** button and the results file will be submitted. The review team will analyze the results and give credit for each item on the official test plan that has been satisfied by this results file.

3.7.3. Automatic Upload from the Local Test Server

This feature is not yet implemented, but it is something we are investigating support for in a future release of the certification tool. For now, please use the manual upload process explained in [Section 3.7.2, “Manual Upload”](#) to submit files from the local test server to the hardware catalog.

3.8. Monitoring Certification Progress

After you submit your results, the review team will analyze their contents and award credit for each passing test that is part of the plan. The review team checks the **Confirmed** box as they verify each passing test, which causes completed sections of the test plan to turn green. 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 your cert request with a question for you, which will automatically be emailed to the person who submitted the cert.

Here is an example of an in-progress test plan where the first two tests were confirmed using data submitted in this certification request, the next test was confirmed using leveraging, and the last test is yet to be confirmed:

NVIDIA Quadro 600 1GB Graphics Card		Confirmed
VIDEO	[250017] (REVIEWED) nVidia Corporation GF108 [Qua▼	Results Hardware (INFO) Results Hardware <input checked="" type="checkbox"/>
NVIDIA Quadro 6000 6GB Graphics Card		Confirmed
VIDEO	[250025] (REVIEWED) nVidia Corporation GF100 [Qua▼	Results Hardware (INFO) Results Hardware <input checked="" type="checkbox"/>
NVIDIA Quadro NVS 450 512 MB PCIe Graphics Card		Confirmed
VIDEO	Leverage from System Certification ▼	Test ID: 208575 Original Certification: 123456 <input checked="" type="checkbox"/>
NVIDIA Tesla C2075 Compute Processor		Confirmed
VIDEO	----- ▼	<input type="checkbox"/>

Figure 3.1. In-Progress Test Plan

When the review team determines that a test has passing results, they use the dropdown box under each test plan entry to select the specific run of the test that contains the passing result. The test ID, which is the number in brackets at the beginning of each line, allows us to track the exact run of the

test that was used to pass this particular test plan entry. When the review team selects a test run for a plan entry, several other pieces of information also appear on the line. These can be seen in the picture above:

- **Results** - The logfiles from the specific test that was run on this device
- **Hardware** - Information on the hardware that was tested, taken from the udev database
- **INFO line Results** - The results of the INFO test that was run at the same time as the test that is being used to satisfy this test plan item
- **INFO line Hardware** - The hardware information from the **info** test that was run at the same time as the test that is being used to satisfy this test plan item. It will have information about every piece of hardware detected in the system, rather than just information on the piece of hardware tested in this run. This is also taken from the udev database.

3.9. Completing Certifications

A certification is complete once all the items on the official test plan have been reviewed and found to have passing results. At this point the certification can be closed and published, or closed and left unpublished. Supplemental certifications always remain unpublished, and system or component certs can also be closed and left unpublished if the vendor does not want to publicly advertise the certification status or the existence of the system/component (most certifications are closed and published).

The lengthy discussions and the system information that is seen by you and the review team will not be visible to the general public in a published cert. All that customers can see when viewing published certs is basic information about the system, including the links that lead to your website. We decided on a basic display to avoid having to constantly update the cert entries when the hardware sold with a particular system changes. All that information is handled by you and your company, and we simply link to it with the specs URL. Here is an example of a completed, published cert as seen by a customer:

Red Hat Hardware Catalog

Hardware Information

Vendor	Mycompany
Make	Ultrasever
Model	X16

Hardware Product Links

Specification URL	http://www.mycompany.com/specs/ultraservers/X16
Product Home URL	http://www.mycompany.com/ultraservers/X16
Product Support URL	http://support.mycompany.com/ultraservers/X16

Certification Information

Category	Server
Product	Red Hat Enterprise Linux
Version	RHEL6.1
Platforms	x86_64
Status	Certified (Supported)

Figure 3.2. Public View of Completed Certification

Appendix A. Appendixes

A.1. Test Procedures

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

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

What the test does: This section explains what the test scripts do. Remember, all the tests are python scripts and can be viewed in the directory `/usr/share/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 to make sure you're running the proper RHEL 5 kernel if you are performing certain tests with more than 32GB of RAM.

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

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

A.1.1. NEW - 1GigEthernet (RHEL 6 only)

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

What the test does: This is a new test that adds link speed detection to the existing **network** test. In addition to passing all the existing network test items, this test must detect a connection speed of 1Gb/s in order to pass. Please see [Section A.1.23, "network"](#) for information on the rest of the test functionality.

A.1.2. NEW - 10GigEthernet (RHEL 6 Only)

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

What the test does: This is a new test that adds link speed detection to the existing **network** test. In addition to passing all the existing network test items, this test must detect a connection speed of 10Gb/s in order to pass. Please see [Section A.1.23, "network"](#) for information on the rest of the test functionality.

A.1.3. NEW - 100MegEthernet (RHEL 6 Only)

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

What the test does: This is a new test that adds link speed detection to the existing **network** test. In addition to passing all the existing network test items, this test must detect a link speed of 100Mb/s in order to pass. Please see [Section A.1.23, "network"](#) for information on the rest of the test functionality.

A.1.4. audio

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

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

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

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

Preparing for the test: Before you begin your test run, you should ensure that the audio test is scheduled and that the system can play and record sound. You can confirm that the test is scheduled by looking for "audio" in the output of the **hwcert** command. 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 tested in these ways:

1. In RHEL 6, right-click on the volume icon at the top of the GUI window and choose **Sound Preferences**. With the tool open, click on the **Hardware** tab, select the sound card you wish to test, then click the **Test Speakers** button. In the window that appears, click the test buttons to generate sounds.
2. In RHEL 5, choose **System** → **Preferences** → **Sound**. With the sound preferences open, click on the **Devices** tab and click the **Test** button under the **Sound Events**, **Sound Playback** item.

If no sound can be heard, ensure that the speakers are plugged in to the correct port. You can use any line-out or headphone jack (we have no requirement for which port you must use). Make sure sound is not muted and try adjusting the volume on the speakers and in the operating system itself. Use the volume control applet in GUI mode (the speaker icon in the top right corner of the desktop) to adjust the master volume higher or lower until you can clearly hear the sound. If that does not work, enter the **Sound Preferences** application (RHEL 6) or the **Volume Control** application (RHEL 5) and try adjusting the **Master**, **PCM**, or **Headphone** sliders. The RHEL 5 Volume Control application can be accessed by right-clicking on the volume icon in the top right-hand corner of the GUI and selecting **Open Volume Control**.

If the audio device has record capabilities, these should also be tested before attempting to run the **hwcert** application. Plug a microphone into one of the Line-in or Mic jacks on the system, or you can use the built-in microphone if you are testing a laptop. Again, we don't require you to use a specific input jack; as long as one works, the test will pass. Use the GUI volume control applet to unmute the correct fader and increase its volume until you can clearly hear sound from the microphone through the speakers (blow on the mic or tap it and listen for the noise through the speakers). You may need to go into the **Edit** → **Preferences** menu in the GUI applet and select additional tracks to be made visible before you can find the one that controls your microphone. The most common one that needs to be made visible is **Microphone**. Once you can hear sound from the microphone through the speakers, switch to the **Capture** screen and unmute the capture slider. Adjust the capture slider to about 75% and test recording sound by using this command:

```
# rec -r 44100 testfile.wav
```

This will record all sounds heard by the mic to a WAV file called **testfile.wav**. Play the file back with the command **play testfile.wav**. If you hear the sound that you recorded, your system is set up to run the audio test. If you do not hear anything, you may need to increase the volume on the capture slider or you may need to adjust additional capture sliders. This may entail using the **Edit** → **Preferences** menu in the GUI volume applet while on the **Capture** tab to make the additional capture sliders visible.

Contact your TAM if you are unable to either play or record sound, as this will lead to a failure of the audio test. If you are able to successfully play and record sound, continue to the next section to learn how to run the test.

Executing the test: The audio test is interactive. Before you execute a run of the hwcert application 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. Execute the hwcert command to begin testing and follow the steps below:

1. When the test reaches the audio portion, you will be given another chance to adjust the volume(s) and make the necessary cable connections. Enter *y* when you are ready to proceed.
2. 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.
3. The system will next play back the file it recorded. If you heard the sound, answer *y* when prompted. Otherwise, answer *n*.

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

A.1.5. battery

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

```
POWER_SUPPLY_TYPE=Battery
```

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

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

Executing the test: The battery test is interactive. Run **hwcert** and schedule the test if it is not already scheduled. When the test runs, it will display the current status of the battery (capacity and

charging status) and ask for the AC adapter to be unplugged until the battery discharges for 10 mWh. The test will automatically end at that point and the tester should plug the AC adapter back in.

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

A.1.6. bluray



Important

Optical drive support for Blu-Ray media may not be detected in RHEL 5. To work around this problem, manually add the **bluray** test with the command **hwcert-backend plan --add --test=bluray --device=\$DEVICE** where **\$DEVICE** is the device you see associated with the other optical drive tests (cdrom and dvd) in the output of the **hwcert** command. You may also need to specify the type of disc, as the system may only be able to detect basic bd (non-writable) support. If you see **Would you like to test bd media (y|n)?** and your system is capable of writing BD-R or BD-RE discs, answer **n** and then enter either **bd-r** to test BD-R discs or **bd-re** to test BD-RE discs.

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

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

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

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

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

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

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

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

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

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

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

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

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

Only formats supported by the drive are scheduled for testing.

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

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

Appendix A. Appendixes

Table A.1. Blank Table of Optical Drive Features

	Rewrite or Write						Read Only		
	BD-RE	DVD+/-RW	CD-RW	BD-R	DVD+/-R	CD-R	BD-ROM	DVD-ROM	CD-ROM
Drive 1									
Drive 2									
Drive 3									
...									
Drive X									

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

Here's an example.

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

Table A.2. Sample Table of Optical Drive Features

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

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

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

disc ejects between each phase (blank, write, compare). The tester will need to insert the disc back into the drive if the drive is not capable of closing the tray by itself, or if it is a slot loading drive.

Executing the test: The bluray test is interactive. To begin testing, install the proper drive as determined by the table you created and schedule the test to be run by the **hwcert** command, if it is not already scheduled. Follow the directions on screen and choose the proper disc format when prompted (the one corresponding with the leftmost column in the table that has an "X" in it), then insert the correct disc when asked. As the test enters the various phases (blank, write, compare, where applicable), the on-screen display will explain what is happening.

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

A.1.7. cdrom

CD drives of all kinds are tested using the same procedures as Blu-Ray drives. Please see [Section A.1.6, "bluray"](#) for more information.

A.1.8. core

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

What the test does: The **core** test is actually composed of two separate routines. The first test is designed to detect clock jitter. Jitter is a condition that occurs when the system clocks are out of sync with each other. The system clocks are not the same as the CPU clock speed, which is just another way to refer to the speed at which the CPUs are operating. The jitter test uses the **gettimeofday()** function to obtain the time as observed by each logical CPU and then analyzes the returned values. If all the CPU clocks are within .2 nanoseconds of each other, the test passes. The tolerances for the jitter test are very tight. In order to get good results it's important that the hwcert 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 hwcert 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 hwcert 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.



Important

If you are testing a system on RHEL 5 with more than 32 CPUs, this test will need to be run on both the bare metal kernel and the Xen kernel. Dom0 in the Xen kernel is only capable of enumerating 32 CPUs, so the bare metal kernel is also required to see CPUs greater than 32. This is only an issue in RHEL 5.

Executing the test: The core test is non-interactive. Run the `hwcert` command with the test scheduled to perform the test.

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

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



Note

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

A.1.9. cpuscaling

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

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

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

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

The test procedure for each CPU package is as follows:

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

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

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

1. The test records the maximum and minimum processor speeds from the file **/sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies**.
2. The userspace governor is selected and maximum frequency is chosen.
3. Maximum speed is confirmed by reading all processors' **/sys/devices/system/cpu/cpuX/cpufreq/scaling_cur_freq** value. If this value does not match the selected frequency, the test will report a failure.
4. Every processor in the package is given the simultaneous task of calculating pi to 2×10^{12} digits. The value for the pi calculation was chosen because it takes a meaningful amount of time to complete (about 30 seconds).
5. The amount of time it took to calculate pi is recorded for each CPU, and an average is calculated for the package.
6. The userspace governor is selected and the minimum speed is set.
7. Minimum speed is confirmed by sysfs data, with a failure occurring if any CPU is not at the requested speed.
8. The same pi calculation is performed by every processor in the package and the results recorded.
9. The ondemand governor is chosen, which throttles the CPU between minimum and maximum speeds depending on workload.
10. Minimum speed is confirmed by sysfs data, with a failure occurring if any CPU is not at the requested speed.
11. The same pi calculation is performed by every processor in the package and the results recorded.
12. The performance governor is chosen, which forces the CPU to maximum speed at all times.
13. Maximum speed is confirmed by sysfs data, with a failure occurring if any CPU is not at the requested speed.
14. The same pi calculation is performed by every processor processor and the results recorded.

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

example, this means that the minimum speed run could take between 54 and 66 seconds and still be considered a passing test (90% of 60 = 54 and 110% of 60 = 66).

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

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

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

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



Important

If you are testing a system on RHEL 5 with more than 32 CPUs, this test will need to be run on the bare metal kernel instead of the Xen kernel. Dom0 in the Xen kernel is only capable of enumerating 32 CPUs, so the bare metal kernel is required to see more than 32. Again, this is only a consideration for RHEL 5.

Executing the test: The cpuscaling test is non-interactive. Run `hwcert` and schedule the test if it is not already scheduled.

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 RHEL 6.4, x86_64. Systems with higher core counts and more populated sockets will take longer. The required info test will add about a minute to the overall run time.

A.1.10. dvd

DVD drives of all kinds are tested using the same procedures as Blu-Ray drives. Please see [Section A.1.6, “bluray”](#) for more information.

A.1.11. NEW - Ethernet (RHEL 6 Only)

What the test covers: The **Ethernet** test only appears when the speed of a network device was 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.

In the local test plan excerpt below, you can see entries for two gigabit Ethernet connections, eth0 and eth1. Eth0 is properly connected to a switch and has the expected 1GigEthernet test planned, but the cable has been disconnected from eth1 which has caused the Ethernet test to be planned for it.

```
7      1GigEthernet eth0      pci0000:00/0000:00:19.0/net/eth0
8      Ethernet   eth1      pci0000:00/0000:00:1c.5/0000:09:00.0/net/eth1
```

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 causes the **Ethernet** test to be planned.

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

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

Advertised pause frame use: No
Advertised auto-negotiation: Yes
Speed: Unknown!
Duplex: Unknown! (255)
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
MDI-X: Unknown
Supports Wake-on: pumbg
```

```
Wake-on: g
Current message level: 0x00000007 (7)
      drv probe link
Link detected: no
```

A.1.12. 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.

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

A.1.13. fencing (Optional)

What the test covers: The **fencing** test covers integrated power control through a baseboard management controller (BMC) or other on-board management system (DRAC, iLO, ILOM etc.) capable of issuing a power cycle command. It is not required for certification at this time.

What the test does: The `hwcert` server reboots the SUT via the SUT's fencing device.

Preparing for the test: The test must be manually added to the test plan on the SUT, as the `hwcert` tool has no way to know if the system has a BMC. To add the test, run the command `hwcert-backend plan --add --test=fencing` and answer `y` to the question of whether or not to add a test without a specific device. The clustering packages that contain the fencing scripts must be installed on the network / `kdump` test server as it is the computer that executes the fencing command. SELinux must also be disabled on the server in order for the test to proceed. Use the command `setenforce 0` on the server to disable SELinux temporarily, or edit `/etc/selinux/config`, set the `SELINUX` parameter to `disabled` and restart the network test server to make the change permanent.

Executing the test: The fencing test is interactive. Run **hwcert** and schedule the test if it is not already scheduled. The test requires a server to be specified, otherwise it will exit with an error, so be sure to supply a local network/kdump test server when prompted. The first thing the test will ask for is the fence agent type (ilo, ipmilan, drac, etc.), followed by the fence device hostname or IP, the username and password. It will then send the fence agent commands to the network/kdump server and the server will execute them on the BMC. The SUT will wait 60 seconds for the reboot to occur. If it does, when it comes back up it queries the hwcert server to determine how long it took to reboot. If the reboot does not occur (the 60 second timeout period on the SUT ends with nothing occurring), the test will fail.

A.1.14. fv_core

The **fv_core** test is a wrapper that launches the FV guest and runs a **core** test on it. Please see [Section A.1.8, “core”](#) for information on the test methodology and run times.

A.1.15. fv_memory

The **fv_memory** test is a wrapper that launches the FV guest and runs a **memory** test on it. Please see [Section A.1.22, “memory”](#) for information on the test methodology and run times.

A.1.16. fv_network (Optional)

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

A.1.17. fv_storage (Optional)

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

A.1.18. hwlatdetect



Note

This test only runs when certifying hardware on the Red Hat Enterprise MRG Realtime product.

What the test covers: The **hwlatdetect** test attempts to find execution delays induced by switches to System Management Mode (SMM). These delays cannot be prevented by the operating system, since they are triggered by a System Management Interrupt (SMI) which cannot be handled by the operating system.

What the test does: The test runs a function under the **stop_machine()** kernel function. The **stop_machine()** function ensures that no kernel activity is happening while it is run (interrupts disabled, preemption disabled, etc). The function being run samples the system clock (generally the Time Stamp Counter (TSC) register) and looks for gaps above a certain threshold. Gaps in consecutive TSC reads mean that the system was interrupted between the reads and executed other code, usually SMM code defined by the system BIOS.

Preparing for the test: Install and boot the realtime kernel before running the hwcert command. It will detect that the running kernel is a realtime kernel and will schedule the hwlatdetect test to be run.

Executing the test: The `hwlatdetect` test is non-interactive. The `hwcert` tool will schedule and run the test automatically when it is run on a realtime kernel.

Run time: The `hwlatdetect` test runs for two hours on all systems. The required info test will add about a minute to the overall run time.

A.1.19. info

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

1. Confirm that `/proc/sys/kernel/tainted` is zero, indicating a non-tainted kernel.
2. Confirm that package verification with `rpm -V` shows that no files have been modified.
3. Confirm that `rpm -qa kernel` shows that the buildhost of the kernel package is a redhat.com machine.
4. Record the boot parameters from `/proc/cmdline` for later analysis by our review team.
5. Confirm that `rpm -V hwcert-client` shows that no modifications have been made to any of the **hwcert** files.
6. Confirm that all the modules shown by `lsmod` show up in a listing of the kernel files with the command `rpm -ql kernel`.
7. Confirm that all modules are on the kABI whitelist.
8. Confirm that the module vendor and buildhost are appropriate Red Hat entries.
9. Confirm that the kernel is the GA kernel of the Red Hat minor release. The verification is attempted with data from the **hwcert-info** package. Internet verification (direct routing/dns resolution have to work, or environment variable 'ftp_proxy=http://proxy.domain:80' has to be set) is attempted if the kernel is not present in the **hwcert-info** package.

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

Run time: The info test takes around 1 minute on a 2013-era, single CPU, 3.3GHz, 6-core/12-thread Intel workstation with 8 GB of RAM running RHEL 6.4, x86_64 that was installed using the kickstart files in this guide. The time will vary depending on the speed of the machine and the quantity of RPM files that are installed.

A.1.20. kdump

What the test covers: The **kdump** test verifies the ability of a system to capture a vmcore after a crash using the **kdump** utility. It tests both local core file storage and the remote copying of a vmcore via NFS to the hwcert test server.

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

Preparing for the test: Ensure that **kdump** is active on the system and specify the DNS name or IP address of the hwcert server when prompted (this is the same server that is used for the network tests) by the hwcert tool. The local kdump test will observe the path parameter in `/etc/kdump.conf`, and default to `/var/crash`. It will also remove any setting of the **net** parameter in that file to force kdump to write the vmcore file locally. The NFS kdump test will set the **net** parameter to the hwcert test server: `hwcert_server:/var/www/hwcert/export` and will observe the path setting also.

Executing the test: The kdump test is interactive. Run **hwcert** and schedule the test if it is not already scheduled. The system will ask you to enter *y* or *n* to trigger the series of crashes when the kdump test is run. The discs will sync and the **vmcore** file will be saved. After the core is saved, the system will reboot and the hwcert application will continue the test automatically, verifying the **vmcore** file is present. It will then repeat the crash, this time exporting the **vmcore** file to the hwcert server system. After the core is exported, the system will reboot and hwcert will continue with any other tests.

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

A.1.21. lid

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

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

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

Preparing for the test: To prepare for the test, ensure that the power management settings do not put the system to sleep or into hibernation when the lid is closed. Right-click on the battery icon in the panel and choose **Preferences**. On the **AC Power** tab, select *Blank screen* as the action that occurs when the lid is closed. Make sure the lid is open before you start the test run.

Executing the test: The lid test is interactive. Run **hwcert** and schedule the test if it is not already scheduled. You will be asked if you are ready to begin the test, so answer *y* 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 *y* if the backlight turns off or *n* if the backlight does not turn off.

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

A.1.22. memory

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

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

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

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

Preparing for the test: Install the correct amount of RAM in the system in accordance with the rules in the Policy Guide. If you are certifying on RHEL 5 with more than 32GB of RAM, please continue reading this section. If you are not certifying on RHEL 5 or you are using RHEL 5 with 32GB of RAM or less, please skip ahead to the "Executing the test" section.

In RHEL 5.3 the default memory limit for dom0 was set to 32GB. This limit was put in place to prevent a memory exhaustion problem that can cause the system to panic on boot. You can read more about the issue in this Red Hat Knowledgebase article: <https://access.redhat.com/site/articles/15694>. Because of that limit, additional testing or a parameter adjustment is necessary to test all of the system RAM. This can be accomplished in either of these two ways:

1. Run the memory test on the RHEL 5 bare metal kernel and the RHEL 5 Xen kernel. The Xen kernel test ensures that 32GB of RAM is addressable in Xen and the bare metal test ensures that all the memory in the system can be seen.
2. Use the **dom0_mem=** parameter in RHEL 5's **grub.conf** file to increase the amount of memory that can be seen in dom0, enabling you to test all the memory inside the Xen kernel and run the memory test only once. This is what the parameter would look like on a RHEL5.3 system with 64GB of RAM:

```
title Red Hat Enterprise Linux Server (2.6.18-128.el5xen)
  root (hd0,0)
  kernel /xen.gz-2.6.18-128.el5 dom0_mem=64G
  module /vmlinuz-2.6.18-128.el5xen ro root=/dev/VolGroup00/LogVol00
  module /initrd-2.6.18-128.el5xen.img
```

Executing the test: The memory test is non-interactive. Run **hwcert** and schedule the test if it is not already scheduled.

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 RHEL 6.4, x86_64. The test will take longer on systems with more RAM. The required info test will add about a minute to the overall run time.

Run time, full-virt guest: The **fv_memory** test takes slightly longer than the bare-metal version, about 18 minutes, to run in a guest. The added time is due to guest startup/shutdown activities and the

required info test that runs in the guest. The required info test on the bare-metal system will add about a minute to the overall run time. The `fv_memory` test run times will not vary as widely from machine to machine as the bare-metal memory tests, as the amount of RAM assigned to our pre-built guest is always the same. There will be variations caused by the speed of the underlying real system, but the amount of RAM in use during the test won't change from machine to machine.



Note

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

A.1.23. network

What the test covers: The **network** test is used to test devices whose function is transferring data over a network. This includes wired Ethernet cards, wireless Ethernet cards and InfiniBand cards if IB is being used as a network protocol. The test will appear as **network** in RHEL 5 test plans and for non-Ethernet devices in RHEL 6. In RHEL 6, testers will see the test listed under different names for Ethernet or Wi-Fi devices:

- **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.
- **100MegEthernet** - The network test with added speed detection for 100 megabit Ethernet connections.
- **WirelessG** - The network test with added speed detection for 802.11g wireless Ethernet connections.
- **WirelessN** - The network test with added speed detection for 802.11n wireless Ethernet connections.

RHEL 6 testers will no longer see a separate **network** test for regular Ethernet or Wi-Fi devices.



Note

If you see a test named **Ethernet** in your local test plan, that is an indication that the test suite did not recognize the speed for that device. Please check the connection and re-run **hwcert** before attempting to test that particular device. See [Section A.1.11, "NEW - Ethernet \(RHEL 6 Only\)"](#) for more information.

What the test does: The first thing the test does is to make sure that a test server was entered when the `hwcert` command was invoked. When that has been located, the test gathers information on all the network devices and runs this procedure:

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

Preparing for testing wired devices: Install as many network devices from the official test plan as you wish. You can run them all at once or one at a time or any combination thereof. Connect each device at its native (maximum) speed or the test will fail. Ensure that the hwcert network test server is up and running. You may specify network addresses for the devices manually or use a DHCP server, but one of those options must be used for each network device.

Preparing for testing wireless Ethernet devices on RHEL 6: In RHEL 6, any system with a supported wireless card will automatically receive any necessary firmware package(s) at install time and all configuration of the cards can be done with the NetworkManager graphical tool. Simply select an SSID on a test network that does not require any additional user input during up/down operations (no authentication requests, VPN login, etc.) and you can run the test as explained in the "Executing the test" section below. In RHEL 5, there are some additional steps that are required. We will discuss them next.

Preparing for testing wireless Ethernet devices on RHEL 5: Most wireless network cards have a closed-source firmware package that is required to make them work. If you try to use the card without the package being installed, an error will appear similar to this one:

```
#ifup wlan0

Determining IP information for wlan0...SIOCSIFFLAGS: No such file or directory
SIOCSIFFLAGS: No such file or directory
failed.
```

In RHEL 5, firmware packages are in a different channel on RHN called the Supplementary channel and they require manual installation. Here are two examples of wireless cards and their associated firmware packages:

- Intel 5000-series - iwI5000-firmware-8.24.2.12-3.el5.noarch

- Intel 6000-series - iwl6000-firmware-9.176.4.1-2.el5.noarch

Even without the firmware installed on RHEL 5, **lspci** will detect the card and the **ifcfg-wlanX** file will be created, but the card cannot work if the firmware is not present. The firmware should be installed in RHEL 5 with the **rpm -ivh packagename** command once it has been downloaded from RHN, and then the computer should be rebooted for the card to be detected properly. Testers should contact their Red Hat Support representative if they need assistance obtaining the firmware package for their wireless cards.

Now that the card is fully operational, it's time to test it. The network test does not use the graphical NetworkManager tool that most users employ to manage their wireless card settings. It uses **ifup** and **ifdown** commands to activate and deactivate network devices in preparation for the test. This means that all configuration parameters for the card, including the wireless network name and any encryption key(s), must be present in the **/etc/sysconfig/network-scripts/ifcfg-wlanX** file. We do not require testing with any specific encryption standard, so you can use no encryption, WEP or WPA/WPA2. Here's an example **ifcfg-wlan0** configuration file that shows all the parameters necessary to connect to a WEP encrypted network:

```
DEVICE=wlan0
BOOTPROTO=dhcp
ONBOOT=no
HWADDR=your hardware address displayed here
ESSID=your network name
KEY=your WEP key
```

Simply place the name of the test network in the ESSID field and the encryption key in the KEY field and bring up the network connection. It should look something like this:

```
# ifup wlan0

Determining IP information for wlan0... done.

# ifconfig wlan0
wlan0      Link encap:Ethernet  HWaddr 00:11:22:33:44:55
            inet addr:192.168.1.104  Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fe80::216:eaff:fe87:ee0a/64  Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

WPA/WPA2 networks require a bit more work. Edit the **/etc/wpa_supplicant/wpa_supplicant.conf** file and enter the appropriate information. This example uses WPA2:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel

network={
    ssid=your network name
    psk=your wpa2 key
    proto=WPA2
    key_mgmt=WPA-PSK
    pairwise=CCMP
    group=CCMP
    priority=1
}
```

Run the command **service wpa_supplicant restart** to (re)start the WPA supplicant daemon.

Edit the **/etc/sysconfig/network-scripts/ifcfg-wlanX** file to include this information:

```
DEVICE=wlan0
BOOTPROTO=dhcp
ONBOOT=no
HWADDR=your hardware address displayed here
```

Then you can run the **ifup wlanX** command to test the wireless configuration. You can begin testing once you achieve a successful connection.

Executing the test: The network test is non-interactive. Run **hwcert** and schedule the test if it is not already scheduled.

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

A.1.24. pccard

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

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

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

Executing the test: The pccard test is non-interactive. Run **hwcert** and schedule the test if it is not already scheduled.

A.1.25. profiler

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

```
options oprofile timer=1
```

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

Executing the test: The profiler test is non-interactive. Run **hwcert** and schedule the test if it is not already scheduled.

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

A.1.26. reboot (Optional)

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

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

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

Executing the test: The reboot test is interactive. You will be asked **Ready to restart?** when you reach the reboot portion of the hwcert program. Answer **y** if you are ready to perform the test. The system will reboot and after coming back up, hwcert will verify that the reboot completed successfully.

A.1.27. rteval



Note

This test only runs when certifying hardware on the Red Hat Enterprise MRG Realtime product.

What the test covers: The **rteval** test is only executed when the kernel that is running is one that has been compiled with realtime extensions (PREEMPT_RT or PREEMPT_RT_FULL). It measures the inherent delay (latency) in servicing timer events, which is a good yardstick for the built-in latency of a particular kernel/hardware combination.

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

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

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

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

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

Preparing for the test: Install and boot the realtime kernel before running **hwcert**. The command will detect that the running kernel is a realtime kernel and will schedule the rteval test to be run.

Running the test: The rteval test is non-interactive. The **hwcert** command will run the test automatically when its scheduling logic is run on a realtime kernel.

Run time: The rteval test runs for 12 hours on all machines. The required info test will add about a minute to the overall run time.

A.1.28. 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, as well as SD media, xD media, MemoryStick and MMC cards. The hwcert test plan script reads through the udev database and looks for storage devices that meet the above criteria. When it finds one, it records the device and its parent and compares it to the parents of any other recorded devices. It does this to ensure that only devices with unique parents are tested. If the parent has not been seen before, the device is added to the test plan. This speeds up testing as only one device per controller will be tested, as per the Policy Guide.

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

1. The script looks through the partition table to locate a swap partition that is not on an LVM or software RAID device. If found, it will deactivate it with **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 **dt** is used to test the device. The dt command is the "data test" program and is a generic test tool capable of testing reads and writes to devices (among other things).
4. After the mounted filesystem test, the filesystem is unmounted and a dt test is performed against the block device, ignoring the file system. The dt test uses the "direct" parameter to handle this.

Preparing for the test: You should install all the drives and storage controllers that are listed on the official test plan. In the case of multiple storage options, as many as can fit into the system at one time can be tested in a single run, or each storage device can be installed individually and have its own run of the storage test. The order of testing and number of controllers present for each test is left up to you. 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, for example, a 2GB swap file placed on the drive.

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 interactive. Run **hwcert** and schedule the test if it is not already scheduled. You will be prompted to choose the disk or partition to be tested when the test executes. If you know the name of the swap partition on a disk that has partitions on it, specify that (the output of **fdisk -l** will tell you the partition types), or enter the name of the disk itself (sda, sdb, sdc, etc.) to have the hwcert application detect the active swap partition for you. If you are testing a blank disk, enter the name of the disk.

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

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

A.1.29. suspend

What the test covers: The **suspend** test covers suspend/resume from S3 sleep state (suspend to RAM) and suspend/resume from S4 hibernation (suspend to disk). This test is only scheduled on systems with a battery (laptops) so it won't be present on any other type of system.

What the test does: The test queries the **/sys/power/state** file and determines which states are supported by the hardware. If it sees "mem" in the file, it schedules the S3 sleep test. If it sees "disk" in the file, it schedules the S4 hibernation test. If it sees both, it schedules both. What follows is the procedure for a system that supports both S3 and S4 states. If your system does not support both types it will only run the tests related to the supported type.

- If S3 sleep is supported, the script uses the **pm-suspend** command to suspend to RAM. The tester wakes the system up after it sleeps and the scripts check the exit code of **pm-suspend** to verify that the system woke up correctly.
- 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.
- If S3 sleep is supported, the tester is prompted to press the key that manually invokes it (a **Fn+F-key** combination or dedicated **Sleep** key). 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.
- If S4 hibernation is supported, the tester is prompted to press the key that manually invokes it (a **Fn+F-key** combination or dedicated **Hibernate** key). 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.

The ability to suspend to RAM is an essential characteristic of laptops. We therefore schedule an automated **pm-suspend** S3 suspend/resume test at the beginning of all hwcert test runs on a laptop. This ensures that all hardware functions normally post-resume. The test will always run on a laptop, much like the info test, regardless of what tests are scheduled.

Preparing for the test: The suspend test cannot be run under the Xen kernel because it has no support for suspend and hibernate activities. Reboot into the non-Xen kernel before running the test.

Executing the test: The suspend test is interactive. Run **hwcert** and schedule the test if it is not already scheduled. When prompted, answer *y* to put the system into S3 sleep mode, press the power button to wake it up after the sleep operation completes, and enter *y* when asked if the sleep operation was successful. Follow the same procedure for the hibernation test if the system supports suspend to disk. The system will then run the same set of tests again, but this time you will be prompted to press a key or key combination to trigger sleep/hibernate. Press *y* when prompted, then put the system into the requested sleep mode. Enter *y* again if the system resumes correctly from its sleep state. If the system being tested has no dedicated keys or key combination that triggers sleep/hibernate, enter *n* when prompted to run the tests using keyboard keys and indicate the lack of support in a note to the review team. If you are testing a RHEL 5 system, reboot into the Xen kernel before continuing with any further testing.



Important

We recommend running this test individually on RHEL 5 as it must be run on a non-Xen kernel.

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

A.1.30. tape

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

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

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

Executing the test: The tape test is non-interactive. Run **hwcert** and schedule the test if it is not already scheduled.

A.1.31. usb (RHEL 5 Only)

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

What the test does: The purpose of the test is to ensure that all USB ports present in a system function as expected. It asks for the number of available USB ports (minus any that are in use for keyboard/mouse, etc.) and then asks the tester to plug and unplug a device into each port. The test watches for USB 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. We recommend using a simple USB device like a mouse rather than a storage device for the test. The reasoning behind that suggestion is that storage devices like USB memory sticks or hard drives may be automounted when they are plugged in. In the second part of the test, where the device is unplugged, unmounted storage devices may generate errors that interfere with the test results.

Preparing for the test: Count the available USB ports and have a spare USB device (a mouse is recommended) available to use during the test.

Executing the test: The usb test is interactive. Run **hwcert** and schedule the test if it is not already scheduled. When prompted by the system, enter the number of available USB 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 USB 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 usb test takes about 15 seconds per USB port. This includes the time to manually plug in the device, scan the port, unplug the device, and scan the port again. The required info test will add about a minute to the overall run time.

A.1.32. NEW - USB2 (RHEL 6 only)

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

What the test does: This is a new test that combines the functionality of the **usb** test with device speed detection. 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 **hwcert** and schedule the test if it is not already scheduled. When prompted by the system, enter the number of available USB2 ports present on the system. Don't count any that are currently in use by keyboards or mice. The system will ask for the test USB2 device to be plugged into a port and will then pause until the tester presses **y** to continue. The system will then ask for the device to be unplugged and again will pause until the tester presses **y** to continue. These steps repeat for the number of ports that were entered. Note that there is no right or wrong order for testing the ports, but each port must be tested only once.

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

A.1.33. NEW - USB3 (RHEL 6 only)

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

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

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

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

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

A.1.34. video

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

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

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

Executing the test: The video test is non-interactive. Run **hwcert** and schedule the test if it is not already scheduled. You will see the screen go blank, followed by a series of test patterns from the **x11perf** test program. It will return to the desktop or to the virtual terminal screen that the system was on at execution time when the test finishes.

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

A.1.35. NEW - WirelessG (RHEL 6 only)

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

What the test does: This is a new test that combines the existing **wlan** and **network** tests and also adds link speed detection. In addition to passing all the existing network test items, this test must

detect a link speed of 22Mb/s in order to pass. Please see [Section A.1.23, “network”](#) for information on the rest of the test functionality.

A.1.36. NEW - WirelessN (RHEL 6 only)

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

What the test does: This is a new test that combines the existing **wlan** and **network** tests and also adds link speed detection. In addition to passing all the existing network test items, this test must detect a link speed of 100Mb/s in order to pass. Please see [Section A.1.23, “network”](#) for information on the rest of the test functionality.

A.1.37. wlan (RHEL 5 Only)

What the test covers: The **wlan** test covers wireless Ethernet devices. It will appear as **wlan** in RHEL 5 test plans, but it will be run as part of either the **WirelessG** or **WirelessN** tests on RHEL 6. It will not appear separately in RHEL 6.

What the test does: The wlan test logs information about the wireless interface, including **/proc/net/wireless** data and the output of **iwconfig**, to ensure the card is up and running. It does not send/receive data as part of the test; that is handled as part of the network test, discussed earlier.

Preparing for the test: Follow the directions in the "Preparing for testing wireless Ethernet devices" section of [Section A.1.23, “network”](#) to associate the card with a wireless network.

Executing the test: The wlan test is non-interactive. Run **hwcert** and schedule the test if it is not already scheduled.

Appendix B. Revision History

Revision 1.0-0 Mon Aug 12 2013

Gary Case gcase@redhat.com

Documentation updated for new 'hwcert' package and moved to the Red Hat Customer Portal.

Revision 1.1-0 Wed Nov 7 2013

Gary Case gcase@redhat.com

Added explanation for the new Hardware Specification entries.

Added new RHEL 6-only network tests.

Removed old references to virt guest configuration files and changed "v7" symlinks to "hwcert".

Index

F

feedback
 contact information for this manual, viii

H

help
 getting help, vii

