

WHITEPAPER

RED HAT ENTERPRISE LINUX: APPLICATION COMPATIBILITY SPECIFICATION

February 2012

EXECUTIVE SUMMARY

This Red Hat Enterprise Linux: Application Compatibility Specification is intended to educate independent software vendors (ISVs) and customers on Red Hat's policies regarding support of third-party applications across multiple releases of the Red Hat Enterprise Linux platform. ISVs and customer applications can avoid major version lock-in by adhering to this published policy during application development.

This document describes the approved uses of the system application programming and binary interfaces that will ensure compatibility across Red Hat Enterprise Linux releases. It outlines the tiered framework under which applications are considered compatible and not compatible.

INTRODUCTION

This specification seeks to provide stability for applications when new releases are deployed, and therefore focuses primarily on forward compatibility issues. Although backward compatibility is provided when possible, development of new capabilities sometimes makes that impractical. Hence, the assurances in this document represent guidelines and published policy that may sometimes be over-ridden by the objective of providing our customers with the most competitive and capable systems possible.

If clarification of these compatibility policies is required, please see the Red Hat Enterprise Linux 6 Developer Guide¹ for additional details, or contact your Red Hat representative. Furthermore, if a published assurance level for functionality that you require is problematic, or has been breached in the goal of delivering new capabilities, please contact your Red Hat representative – we will try to meet your needs.

¹ http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Developer_Guide/index.html

TERMINOLOGY

The following are basic terms that are used in this document:

- **Binary compatibility**
Binary compatibility specifies whether applications that are compiled on a combination of Red Hat Enterprise Linux and a particular hardware architecture will load and run predictably across different instances of the operating environment. These application binaries include executable files and Dynamic Shared Objects (DSO), and the level of compatibility is defined by a specified Application Binary Interface (ABI).
- **Application Programming Interface (API)**
An API is an interface implemented by a software program that enables it to interact with other software including operating system components. The API is enforced at compile time and determines source compatibility, that is, whether application source code will compile predictably across different instances of the operating environment.
- **Application Binary Interface (ABI)**
An ABI is a set of runtime conventions adhered to by all tools that interact with a compiled binary representation of a program. It is a superset of the API and describes the low-level interface between an application program and the operating environment. It covers details such as:
 - data type, size, and alignment
 - the calling standard, which defines how function arguments are passed and return values retrieved
 - the binary format of object files, program libraries

Examples of such tools include compilers, linkers, runtime libraries, and the operating system itself. An ABI is enforced at run time.

- **ABI conformance**
A compiler conforms to an ABI if it generates code that follows all of the specifications enumerated by that ABI. A library conforms to an ABI if it is implemented according to that ABI. An application conforms to an ABI if it is built using tools that conform to that ABI and does not contain source code that changes behavior specified by the ABI.
- **Core persistent system infrastructure**
The core persistent system infrastructure refers to interfaces and externally available data structures that represent system state or provide a means of communicating with the system (for instance, system calls and header files).
- **Compatibility in a virtualized environment**
Virtual environments emulate bare metal environments such that unprivileged applications that run on bare metal environments will run, unmodified, in corresponding virtual environments. Virtual environments present simplified views of physical resources, so some differences may exist e.g. hardware or machine types and characteristics may be abstractions rather than concrete representations of the underlying system.
- **Dynamic Software Collections**
Dynamic Software Collections are sets of applications closely related by their interactions (e.g., various programs making up the line printer spooling system). Because these collections work as integrated solutions, changes to one portion are more likely to affect others in the collection. Some Dynamic Software Collections change more frequently to provide expanded functionality, and it is generally desirable to make these more rapidly available. These Dynamic Software Collections are on an accelerated life cycle, and so their compatibility assurance is limited.

- **Supplementary packages, Optional packages**

Packages that are not part of the product but are included (via separate channels in Red Hat Network) for convenience or to satisfy build dependencies.

- **Major and minor releases**

A Red Hat major release represents a significant step in the development of a product (revolutionary changes are usually reserved for major releases), and is typically designated by a single numeral (e.g., Red Hat Enterprise Linux 6). Minor releases appear more frequently, always within the scope of a major release, and generally represent smaller, incremental developmental steps.

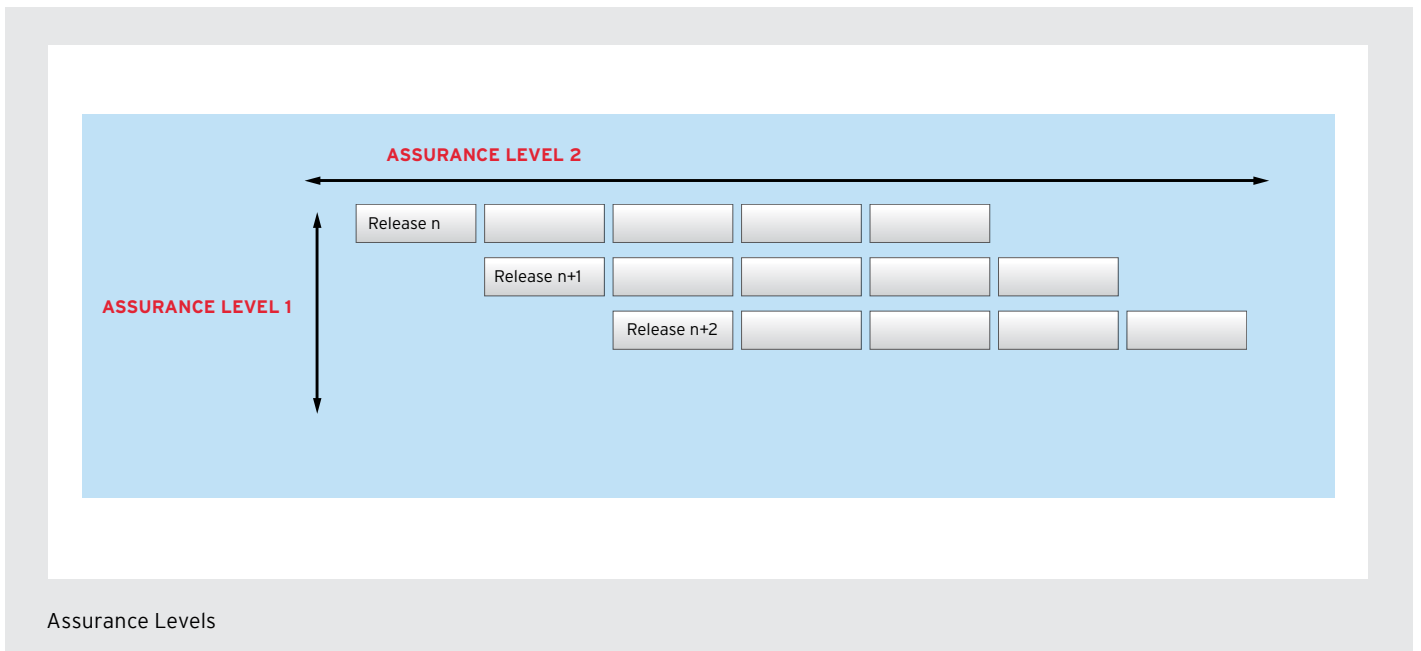
Scope of Compatibility

Packages in Red Hat Enterprise Linux are classified under one of the following four assurance levels:

- **Assurance Level 1:** APIs and ABIs are stable across three major releases, the release that introduces a new or revised API or ABI, and the two following major releases (n, n+1, n+2).

In those cases where changes to a library during its assurance lifetime will cause an incompatibility with existing compiled code, a compatibility library will be provided to allow that code to continue to run without modifications.

- **Assurance Level 2:** APIs and ABIs are stable within one major release
- **Assurance Level 3:** Reserved for future use²
- **Assurance Level 4:** No assurance is provided.



² This assurance level addresses binary compatibility for those customers who have used EUS or Long Life (LL) to extend support of features that have changed over intervening minor releases

For assurance levels for specific Red Hat Enterprise Linux packages, please see the Appendices.

Assurance levels for bare metal configurations apply to virtualized configurations except for any features that directly interact with hardware. Those hardware-related features have no API or ABI assurance. For example, applications that rely on GPU features cannot be assured of binary compatibility.

COMPATIBILITY ASSURANCES

- **API**
Source code will continue to compile into a working application with equivalent functionality.
- **ABI**
Source code compiled into binary form and linked with libraries using public binary interfaces will run with equivalent functionality.

Notes:

- This assurance applies to components explicitly included in the product (that is, directly installable by the installer).
- This assurance does not apply to technology preview components.
- **Configuration Files**
All configuration files are supported at assurance level 2.

Notes:

- Existing configuration files will either remain syntactically and functionally consistent, or will be automatically upgraded to a functionally equivalent form.
- **Application Data Compatibility**
Application data files are stable within assurance level claims.

Notes:

- Existing application data will remain syntactically and functionally consistent.
- When in backwards compatibility mode (which is explicitly invoked by the user), application reads are assured to work within assurance level claims, but writes may not.
- **Core Persistent System Infrastructure**
The core persistent system infrastructure will remain syntactically and functionally consistent within assurance level claims.

This applies to the following infrastructure components:

- Raid data structures
- Volume manager information
- Filesystem labels
- File systems
- Attributes
- RPM file formats and signatures

Packages in Red Hat Enterprise Linux 6 are signed with SHA256 signatures. This makes them incompatible by default with the previous versions of Yum and RPM.

- **Virtualized Environment**
An application that runs on bare metal will run, unmodified, in a guest.
- **Dynamic Software Collections**
Please see Appendix B for assurance levels for specific Dynamic Software Collections.
- **Supplementary Packages and Optional Packages**
Packages that are in the Supplementary and Optional Red Hat Network channels have no API or ABI assurance.

BEST PRACTICES FOR PRESERVING BINARY COMPATIBILITY

Red Hat recommends that application developers adhere to the following to ensure that their applications run predictably without modification:

- 1. Build applications using the public interfaces of a library.** Internal interfaces are subject to change at any time, which can cause instability in the dependent application if improperly linked.
- 2. Avoid static linking of libraries (C/C++).** Static linking causes the executables to have their own version of the library. This increases the chance of an application not running predictably on a later version of the operating system as these library dependencies might have changed along the way. Linking applications dynamically is the preferred alternative to this problem.
- 3. If you require functionality from a package that is currently at a lower level of assurance than you wish, please contact Red Hat.**
- 4. Limit linking applications to the core libraries.** The core libraries (see Appendix A) are assured to preserve binary compatibility across major releases.
- 5. Provide compatibility libraries for applications that have been built on libraries that do not assure binary compatibility at the desired level.** Even in that case the bundled libraries must themselves use only the interfaces provided by the core libraries.
- 6. Package applications using the RPM mechanism.** RPM provides a robust software packaging mechanism that includes rigorous specification of application dependencies. When creating RPMs, the following should be kept in mind:
 - (a) Avoid using RPM triggers whenever possible.
 - (b) Don't depend on the execution order of pre-install or pre-uninstall scripts, which may change between releases.
 - (c) Explicitly state all required runtime and build dependencies using the appropriate RPM syntax.
 - (d) Do not modify, replace, or recompile files managed by Red Hat-provided RPM packages.
 - (e) When considering dependencies, don't assume that all possible packages will be installed on every Red Hat Enterprise Linux system. The default installed packages may change between releases and between product variants of the same version. For instance, the Workstation product will have a different installed package set than the Server product.

7. Follow the Filesystem Hierarchy Standard (FHS) version 2.3 when installing programs. Third-party software should be installed to the '/opt' subdirectory. More information on the FHS is available at:

<http://www.pathname.com/fhs/>

8. Applications should be built against libraries that correspond to the native hardware environment rather than a compatibility layer on top of the hardware. This is because the compatibility userspace contains a subset of system libraries compared to the native userspace.

9. Do not design applications that rely on configuration files used by system packages. These files can change between major releases unless the upstream community is explicitly committed to preserving them.

APPENDIX A: ASSURANCE LEVELS FOR SPECIFIC PACKAGES AND LIBRARIES

ASSURANCE LEVEL 1: APIS AND ABIS ARE STABLE ACROSS THREE MAJOR RELEASES

alsa-lib	libgcc	libtopology	openmotif
elfutils-libelf	libgfortran	libvirt-client	openssl
glibc	libgomp	libxml2	pam
glibc-utils	libselenium	libxslt	SDL
gtk2	libselenium-python	mesa-libGL	
krb5-libs	libstdc++	mesa-libGLU	

ASSURANCE LEVEL 2: APIS AND ABIS ARE STABLE WITHIN ONE MAJOR RELEASE

atk	gstreamer-plugins-base	libgnome	pcre
audit-libs	httpd	libgnomeui	perl
audit-libs-python	java-1.6.0-openjdk	libgudev1	perl-Digest-SHA
boost-date-time	kdebase	libhugetlbfs	perl-libs
boost-filesystem	kdebase-libs	libICE	perl-Time-Piece
boost-graph	kdebase-workspace	libjpeg	phonon-backend-gstreamer
boost-iostreams	kdebase-workspace-akonadi	libnotify	polkit
boost-math	kdebase-workspace-libs	libpng	popt
boost-program-options	kdegraphics	librsvg2	postgresql-libs
boost-python	kdegraphics-libs	libSM	pulseaudio
boost-regex	kdelibs	libtiff	pulseaudio-libs
boost-serialization	kdemultimedia	libudev	pulseaudio-libs-glib2
boost-signals	kdemultimedia-libs	libusb	PyQt4
boost-system	kdenetwork	libuuid	python
boost-test	kdenetwork-libs	libX11	python-libs
boost-thread	kdepim	libXau	qt
boost-wave	kdepim-libs	libXaw	qt-mysql
bzip2-libs	kdepimlibs	libXext	qt-odbc
cairo	kdepimlibs-akonadi	libXft	qt-postgresql
corosync	kdesdk	libXi	qt-x11
cups-libs	kdesdk-devel	libXinerama	qt3
cyrus-sasl-gssapi	kdesdk-libs	libXmu	qt3-MySQL
cyrus-sasl-lib	kdesdk-utils	libXpm	qt3-ODNC
cyrus-sasl-md5	kdm	libXrandr	qt3-PostgreSQL

ASSURANCE LEVEL 2 CONTINUED...

db4	libacl	libXrender	readline
db4-cxx	libaio	libXt	sqlite
dbus-glib	libattr	libXtst	startup-notification
dbus-libs	libblkid	mysql-libs	systemtap
elfutils-libs	libbonobo	ncurses-libs	tcl
expat	libcanberra	net-snmp-libs	tcp_wrappers-libs
fuse-libs	libcanberra-gtk2	net-snmp-perl	tk
GConf2	libcap-ng	net-snmp-python	unique
glib2	libcurl	nss	xz-libs
gmp	libgcj	nss-sysinit	zlib
gnome-keyring	libgcj-devel	numactl	
gnome-keyring-pam	libgcrypt	pango	
gnutls	libglade2	papi	

ASSURANCE LEVEL 3: RESERVED FOR FUTURE USE

(This section is intentionally left blank)

ASSURANCE LEVEL 4: STABILITY OF APIS OR ABIS SUBJECT TO CHANGE AT RED HAT'S DISCRETION

binutils	junit	openldap	postgresql-test
e2fsprogs-libs	libcgroup	postgresql-contrib	pulseaudio-libs-zeroconf
glade3-libgladeui	libcom_err	postgresql-docs	pulseaudio-module-bluetooth
gnome-desktop	libdrm	postgresql-plperl	pulseaudio-module-gconf
gstreamer	libss	postgresql-plpython	pulseaudio-module-x11
gstreamer-devel	mesa-dri-drivers	postgresql-pltcl	pulseaudio-utils
gvfs	mysql-server	postgresql-server	tkinter

APPENDIX B: ASSURANCE LEVELS FOR DYNAMIC SOFTWARE COLLECTIONS

Red Hat Enterprise Linux 6 includes the following Dynamic Software Collections:

1. LAMP - (Linux), Apache, MySQL, PHP, Perl, Python, and PostgreSQL
2. Java - OpenJDK and tomcat
3. Ruby and associated Gems
4. Printing - cups and samba

Packages within these collections will typically be updated more frequently than the traditional major release refresh cycle. Therefore, Dynamic Software Collections have their own lifecycles, independent of major releases.

Dynamic Software Collections have an assurance level based on releases of the Dynamic Software Collection itself (not the Red Hat Enterprise Linux release containing the Dynamic Software Collection), and are stable for three releases of the Dynamic Software Collection (n, n+1, n+2).

Users will have the option to remain on an older version of an Dynamic Software Collection for up to five years.

Red Hat will add more collections in the future at its discretion based on customer demand.

ABOUT RED HAT

Red Hat was founded in 1993 and is headquartered in Raleigh, NC. Today, with more than 70 offices around the world, Red Hat is the largest publicly traded technology company fully committed to open source. That commitment has paid off over time, for us and our customers, proving the value of open source software and establishing a viable business model built around the open source way.

SALES AND INQUIRIES

NORTH AMERICA
1-888-REDHAT1
www.redhat.com

**EUROPE, MIDDLE EAST
AND AFRICA**
00800 7334 2835
www.europe.redhat.com
europe@redhat.com

ASIA PACIFIC
+65 6490 4200
www.apac.redhat.com
apac@redhat.com

LATIN AMERICA
+54 11 4329 7300
www.latam.redhat.com
info-latam@redhat.com