



<https://access.redhat.com/articles/rhel9-abi-compatibility>

Terminology

The following are basic terms used in this document:

- **Application programming interface (API)**

An API is a public 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 similarly across different versions of the operating environment. It covers details such as:

- C and C++ headers for developer use
- Language syntax, but only where it applies to a published standard
- Public interface definitions
- Command line interfaces, but only where it applies to documented public interfaces

- **Application binary interface (ABI)**

An ABI is a set of runtime conventions that interact with a compiled binary representation of a program. The ABI is enforced at runtime and it describes the low-



<https://access.redhat.com/articles/rhel9-abi-compatibility>

- **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 or that otherwise bypasses the ABI.

- **Binary compatibility**

Binary compatibility means application binaries that are compiled for a specific ABI, generally for a combination of RHEL and a particular hardware architecture, will load and run similarly across different versions of RHEL. Application binaries consist of executable files, Dynamic Shared Objects (DSO), source, bytecode for interpreted just-in-time compiled languages, and their required data files.

- **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