



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

Compatibility levels

All components and packages in Red Hat Enterprise Linux are classified under one of the following four compatibility levels:

- **Compatibility level 1**

- APIs and ABIs are stable within the lifetime of a major release and ABIs are also stable across the next two major releases; the release that introduces a new or revised ABI, and the two following major releases (n , $n+1$, $n+2$). In the case of this document, release n starts with Red Hat Enterprise Linux 9. If a change to a library causes an incompatibility with existing binaries, a separate version of the library will be provided with the older ABI to run the application without modification.

- **Compatibility level 2**

- APIs and ABIs are stable within the lifetime of a single major release. Compatibility level 2 application interfaces will not change from minor release to minor release and can be relied upon by the application to be stable for the duration of the major release. Compatibility level 2 is the default for packages in Red Hat Enterprise Linux 9. Packages not identified as having another compatibility level may be considered compatibility level 2.



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

Compatibility exceptions

The following are exceptions to compatibility in RHEL.

SystemTap static probes

- No assurances are made at this time that integrated SystemTap static probes will continue to have the same probe name, probe location, or interpretation or number of arguments. Since the probes are primarily designed for deep analysis and debugging, the probes must be able to change as the underlying implementation changes.

Static linking with the C/C++ runtime

- Static linking with the C/C runtime is not supported. This includes linking with any files that are part of the ``glibc-static`` or ``libstdc`-static` packages. You may choose to link statically, but the resulting application binaries may fail to operate if any package in the installation is changed.

C/C++ application sanitizers

- C/C++ applications built with the compiler option `-fsanitize=[option]` cannot participate in the API or ABI guarantees provided in this document. The sanitizer