Red Hat Developer Toolset 12

12.0 Release Notes

Release Notes for Red Hat Developer Toolset 12.0

Last Updated: 2023-04-27

Red Hat Developer Toolset 12 12.0 Release Notes

Release Notes for Red Hat Developer Toolset 12.0

Lenka Špačková
Red Hat Customer Content Services
lspackova@redhat.com

Jaromír Hradílek
Red Hat Customer Content Services
jhradilek@redhat.com

Eliška Slobodová
Red Hat Customer Content Services

Legal Notice

Abstract

Red Hat Developer Toolset is a Software Collection from the Red Hat Software Collections offering. It is designed for developers working on the Red Hat Enterprise Linux platform. Using a framework called Software Collections, an additional set of tools is installed into the /opt/ directory, as recommended by the UNIX Filesystem Hierarchy Standard. These tools are enabled by the user on demand using the supplied scl utility. Red Hat Developer Toolset 12.0 provides current versions of the GNU Compiler Collection, GNU Debugger, and other development, debugging, and performance monitoring tools. These do not replace the Red Hat Enterprise Linux system versions of these tools, nor will they be used in preference to those system versions unless explicitly invoked using the scl utility. These Release Notes contain important information available at the time of release of Red Hat Developer Toolset 12.0. Known problems, resources, and other issues are discussed here. Read this document before beginning to use Red Hat Developer Toolset 12.0. For

information on other components of Red Hat Software Collections, see the Red Hat Software Collections Release Notes. For details about length of support for individual components, refer to the Red Hat Software Collections Product Life Cycle document.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

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

CHAPTER 1. SYSTEM REQUIREMENTS

1.1. INSTALLATION PREREQUISITES

Red Hat Developer Toolset 12.0 contains packages for Red Hat Enterprise Linux 7 and supports the following architectures:

- AMD64 and Intel 64
- IBM POWER, big endian
- IBM POWER, little endian
- 64-bit IBM Z

Generation and manipulation of 32-bit binaries is also supported on the AMD64 and Intel 64 architectures.

NOTE

In Red Hat Enterprise Linux 8, **GCC Toolset** is available, which provides tools similar to Red Hat Developer Toolset. For details, see Using GCC Toolset.

Before installing Red Hat Developer Toolset 12.0, it is recommended to apply all available Red Hat Enterprise Linux errata updates. This will enable all Red Hat Developer Toolset features and apply fixes that can otherwise impact the tools or built code.

Red Hat Developer Toolset 12.0 requires systems to be subscribed to the **Optional** repository in order to access all the necessary dependent packages required by the Red Hat Developer Toolset toolchain. Packages from the **Optional** repository are listed in the Red Hat Software Collections 3.8 Release Notes.

If you use Red Hat Subscription Manager, you can add these repositories by navigating to **Applications → System Tools → Red Hat Subscription Manager**.

Then enable the **Optional** repository for your system. For detailed information, see https://access.redhat.com/solutions/392003.

Users of Red Hat Satellite can enable the **Optional** repository by following the instructions at https://access.redhat.com/solutions/70019.

Since Red Hat Developer Toolset 4.1, the content is available also in the ISO format at https://access.redhat.com/downloads, specifically for Server and Workstation.

NOTE

Packages that require the **Optional** repository cannot be installed from the ISO image. A list of packages that require enabling of the **Optional** repository is provided in the Red Hat Software Collections Release Notes.

Beta content is unavailable in the ISO format.

1.2. INSTALLATION AND UPGRADING

Use of Red Hat Developer Toolset 12.0 requires the removal of any earlier pre-release versions, including Beta releases. If you have installed any previous version of Red Hat Developer Toolset, uninstall it from your system as described in the Uninstalling Red Hat Developer Toolset section of the *Red Hat Developer Toolset User Guide*.

Note that installing Red Hat Developer Toolset 12.x in parallel with Red Hat Developer Toolset 11.x is supported.

For detailed information on installation of Red Hat Developer Toolset 12.0 see the Installing Red Hat Developer Toolset chapter of the *Red Hat Developer Toolset User Guide*.

The in-place upgrade from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7 is not supported by Red Hat Developer Toolset. As a consequence, the installed components might not work correctly after the upgrade. If you want to upgrade from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7, it is strongly recommended to remove all Red Hat Developer Toolset packages, perform the in-place upgrade, update the Red Hat Developer Toolset repository, and install the Red Hat Developer Toolset packages again. It is advisable to back up all data before upgrading.

CHAPTER 2. RED HAT DEVELOPER TOOLSET 12.0 RELEASE

2.1. FEATURES

2.1.1. List of Components

Red Hat Developer Toolset 12.0 provides the following components:

Development Tools

- GNU Compiler Collection (GCC)
- binutils
- elfutils
- dwz
- make
- annobin

Debugging Tools

- GNU Debugger (GDB)
- strace
- ltrace
- memstomp

Performance Monitoring Tools

- SystemTap
- Valgrind
- OProfile
- Dyninst

For details, see the Red Hat Developer Toolset Components table in the *Red Hat Developer Toolset User Guide*.

2.1.2. Changes in Red Hat Developer Toolset 12.0

All components in Red Hat Developer Toolset 12.0 are distributed with the **devtoolset-12-** prefix and only for Red Hat Enterprise Linux 7.

The following components have been upgraded in Red Hat Developer Toolset 12.0 compared to the previous release of Red Hat Developer Toolset:

- **GCC** to version 12.1.1

- **annobin** to version 10.76
- **elfutils** to version 0.187
- **GDB** to version 11.2
- **strace** to version 5.18
- **SystemTap** to version 4.7
- **Valgrind** to version 3.19.0
- **Dyninst** to version 12.1.0

In addition, a bug fix update is available for **binutils**.

For detailed information on changes in Red Hat Developer Toolset 12.0, see Red Hat Developer Toolset User Guide.

2.1.3. Container Images

The following container images are available with Red Hat Developer Toolset:

- rhscl/devtoolset-12-perftools-rhel7
- rhscl/devtoolset-12-toolchain-rhel7

For more information, see the Red Hat Developer Toolset Images chapter in *Using Red Hat Software Collections Container Images*.

Note that only the latest version of each container image is supported.

2.2. KNOWN ISSUES

dyninst component, BZ#1763157

Dyninst 12 is provided only for the AMD64 and Intel 64 architectures.

gcc component, BZ#1731555

Executable files created with Red Hat Developer Toolset are dynamically linked in a nonstandard way. As a consequence, Fortran code cannot handle input/output (I/O) operations asynchronously even if this functionality is requested. To work around this problem, link the **libgfortran** library statically with the **-static-libgfortran** option to enable asynchronous I/O operations in Fortran code. Note that Red Hat discourages static linking for security reasons.

gcc component, BZ#1570853

In Red Hat Developer Toolset, libraries are linked via linker scripts which might specify some symbols through static archives. This is required to ensure compatibility with multiple versions of Red Hat Enterprise Linux. However, the linker scripts use names of the respective shared object files. As a consequence, the linker uses different symbol handling rules than expected, and does not recognize symbols required by object files when the option adding the library is specified before options specifying the object files, such as:

```
gcc -lsomelib objfile.o
```

Such use of a library from Red Hat Developer Toolset results in linker error messages **undefined reference to symbol**. To enable successful symbol resolution and linking, follow the standard linking practice and specify the option adding the library after the options specifying the object files:

```
gcc objfile.o -lsomelib
```

Note that this recommendation applies when using the version of **GCC** available as a part of Red Hat Enterprise Linux, too.

gcc component, BZ#1433946

GCC in Red Hat Developer Toolset 3.x contained the libasan package, which might have conflicted with the system version of libasan. As a consequence, depending on which libasan was present in the system, the **-fsanitize=address** tool worked only either with the system **GCC** or with the Red Hat Developer Toolset version of **GCC**, but not with both at the same time. To prevent the described conflict, in Red Hat Developer Toolset 4.x and later versions, the package was renamed to libasan*N*, where *N* is a number. However, if the Red Hat Software Collections repository is enabled, the problem can occur after the system update because the system version of libasan is available in an earlier version than the Red Hat Developer Toolset 3.x version, which is still available in the repository. To work around this problem, exclude this package while updating:

```
~]$ yum update --exclude=libasan
```

oprofile component

OProfile 1.3.0 and **OProfile 1.2.0** shipped in Red Hat Developer Toolset works on all supported architectures, with the exception of IBM Z, where only the **ocount** tool works on the following models: z196, zEC12, and z13. **operf** and the other tools, such as **oparchive** or **opannotate**, do not work on IBM Z. For profiling purposes, users are recommended to use the Red Hat Enterprise Linux 7 system **OProfile 0.9.9** version, which supports **opcontrol** with **TIMER** software interrupts.

Note that for correct reporting of data collected by **OProfile 0.9.9**, the corresponding **opreport** utility is necessary. Thus **opcontrol**-based profiling should be performed with Red Hat Developer Toolset disabled because the reporting tools from Red Hat Developer Toolset cannot process data collected within **opcontrol** legacy mode correctly.

valgrind component, BZ#869184

The default **Valgrind gdbserver** support (***--vgdb=yes***) can cause certain register and flags values to be not always up-to-date due to optimizations done by the **Valgrind** core. The **GDB** utility is therefore unable to show certain parameters or variables of programs running under **Valgrind**. To work around this problem, use the ***--vgdb=full*** parameter. Note that programs might run slower under **Valgrind** when this parameter is used.

multiple components

The devtoolset-*version*-*package_name*-debuginfo packages can conflict with the corresponding packages from the base Red Hat Enterprise Linux system or from other versions of Red Hat Developer Toolset. This namely applies to devtoolset-*version*-gcc-debuginfo, devtoolset-*version*-ltrace-debuginfo, devtoolset-*version*-valgrind-debuginfo, and might apply to other debuginfo packages, too. A similar conflict can also occur in a multilib environment, where 64-bit debuginfo packages conflict with 32-bit debuginfo packages.

For example, on Red Hat Enterprise Linux 7, devtoolset-7-gcc-debuginfo conflicts with three packages: gcc-base-debuginfo, gcc-debuginfo, and gcc-libraries-debuginfo. On Red Hat Enterprise Linux 6, devtoolset-7-gcc-debuginfo conflicts with one package: gcc-libraries-debuginfo.

As a consequence, if conflicting debuginfo packages are installed, attempts to install Red Hat Developer Toolset can fail with a transaction check error message similar to the following examples:

> file /usr/lib/debug/usr/lib64/libitm.so.1.0.0.debug from install of gcc-base-debuginfo-4.8.5-16.el7.x86_64 conflicts with file from package devtoolset-7-gcc-debuginfo-7.2.1-1.el7.x86_64

> file /usr/lib/debug/usr/lib64/libtsan.so.0.0.0.debug from install of gcc-debuginfo-4.8.5-16.el7.x86_64 conflicts with file from package devtoolset-7-gcc-debuginfo-7.2.1-1.el7.x86_64

> file /usr/lib/debug/usr/lib64/libitm.so.1.0.0.debug from install of devtoolset-7-gcc-debuginfo-7.2.1-1.el6.x86_64 conflicts with file from package gcc-libraries-debuginfo-7.1.1-2.3.1.el6_9.x86_64

To work around the problem, manually uninstall the conflicting debuginfo packages prior to installing Red Hat Developer Toolset 12.0. It is advisable to install only the relevant debuginfo packages when necessary and expect such problems to happen.

Other Notes

- Red Hat Developer Toolset primarily aims to provide a compiler for development of user applications for deployment on multiple versions of Red Hat Enterprise Linux. Operating system components, kernel modules and device drivers generally correspond to a specific version of Red Hat Enterprise Linux, for which the supplied base OS compiler is recommended.

- Red Hat Developer Toolset 12.0 supports only C, C++ and Fortran development. For other languages, invoke the system version of **GCC** available on Red Hat Enterprise Linux.

- Building an application with Red Hat Developer Toolset 12.0 on Red Hat Enterprise Linux (for example, Red Hat Enterprise Linux 7) and then executing that application on an earlier minor version (such as Red Hat Enterprise Linux 6.7.z) may result in runtime errors due to differences in non-toolchain components between Red Hat Enterprise Linux releases. Users are advised to check compatibility carefully. Red Hat supports only execution of an application built with Red Hat Developer Toolset on the same, or a later, supported release of Red Hat Enterprise Linux than the version used to build that application.

- **Valgrind** must be rebuilt without Red Hat Developer Toolset's **GCC** installed, or it will be used in preference to Red Hat Enterprise Linux system **GCC**. The binary files shipped by Red Hat are built using the system **GCC**. For any testing, Red Hat Developer Toolset's **GDB** should be used.

- All code in the non-shared library **libstdc++_nonshared.a** in Red Hat Developer Toolset 12.0 is licensed under the GNU General Public License v3 with additional permissions granted under Section 7, described in the GCC Runtime Library Exception version 3.1, as published by the Free Software Foundation.

- The compiler included in Red Hat Developer Toolset emits newer DWARF debugging records than previous compilers available on Red Hat Enterprise Linux. These new debugging records improve the debugging experience in a variety of ways, particularly for C++ and optimized code. However, certain tools are not yet capable of handling the newer DWARF debug records. To generate the older style debugging records, use the options **-gdwarf-2 -gstrict-dwarf** or **-gdwarf-3 -gstrict-dwarf**.

- Some newer library features are statically linked into applications built with Red Hat Developer Toolset to support execution on multiple versions of Red Hat Enterprise Linux. This adds a small additional security risk because regular Red Hat Enterprise Linux errata would not change this code. If the need for developers to rebuild their applications due to such an issue arises, Red Hat will signal this in a security erratum. Developers are strongly advised not to statically link their entire application for the same reasons.

- Note that error messages related to a missing **libitm** library when using the **-fgnu-tm** option require the libitm package to be installed. You can install the package with the following command:

  ```
  yum install libitm
  ```

- To use the **ccache** utility with **GCC** included in Red Hat Developer Toolset, set your environment correctly. For example:

  ```
  ~]$ scl enable devtoolset-12 '/usr/lib64/ccache/gcc -c foo.c'
  ```

  Alternatively, you can create a shell with the Red Hat Developer Toolset version of **GCC** as the default compiler:

  ```
  ~]$ scl enable devtoolset-12 'bash'
  ```

  After you have created the shell, run the following two commands:

  ```
  ~]$ export PATH=/usr/lib64/ccache${PATH:+:${PATH}}
  ```

  ```
  ~]$ gcc -c foo.c
  ```

- Because the **elfutils** libraries contained in Red Hat Developer Toolset 12.0 are linked to a client application statically, caution is advised when passing handles to **libelf**, **libdw**, and **libasm** data structures to external code and when passing handles received from external code to **libelf**, **libdw**, and **libasm**.

  Be especially careful when an external library, which is linked dynamically against the system version of **elfutils**, is passed a pointer to a structure that comes from the Red Hat Developer Toolset 12.0 version of **elfutils** (or vice versa).

  Generally, data structures used in the Red Hat Developer Toolset 12.0 version of **elfutils** are not compatible with the Red Hat Enterprise Linux system versions, and structures coming from one should never be touched by the other.

  In applications that use the Red Hat Developer Toolset 12.0 libraries, all code that was linked against the system version of the libraries should be recompiled against the libraries included in Red Hat Developer Toolset 12.0.

- The **elfutils EBL** library, which is used internally by **libdw**, was amended not to open back ends dynamically. Instead, a selection of back ends is compiled in the library itself: the 32-bit AMD and Intel architecture, AMD64 and Intel 64 systems, Intel Itanium, IBM Z, 32-bit IBM Power Systems, 64-bit IBM Power Systems, IBM POWER, big endian, and the 64-bit ARM architecture. Some functionality may not be available if the client wishes to work with ELF files from architectures other than those mentioned above.

- Some packages managed by the **scl** utility include privileged services that require **sudo**. The system **sudo** clears environment variables and so Red Hat Developer Toolset includes its own **sudo** shell script, wrapping **scl enable**. This script does not currently parse or pass normal **sudo** options, only **sudo *COMMAND ARGS ...***. In order to use the system version of **sudo** from within a Red Hat Developer Toolset-enabled shell, use the **/usr/bin/sudo** binary.

- Intel have issued erratum HSW136 concerning TSX (Transactional Synchronization Extensions) instructions. Under certain circumstances, software using the Intel TSX instructions may result in unpredictable behavior. TSX instructions may be executed by applications built with Red Hat Developer Toolset **GCC** under certain conditions. These include use of **GCC**'s experimental Transactional Memory support (using the **-fgnu-tm** option) when executed on hardware with TSX instructions enabled. The users of Red Hat Developer Toolset are advised to exercise further caution when experimenting with Transaction Memory at this time, or to disable TSX instructions by applying an appropriate hardware or firmware update.

- To use the Memory Protection Extensions (MPX) feature in **GCC**, the Red Hat Developer Toolset version of the **libmpx** library is required, otherwise the application might not link properly.

- The two **binutils** linkers, **gold** and **ld**, have different ways of handling hidden symbols, which leads to incompatibilities in their behavior. Previously, the **gold** and **ld** linkers had inconsistent and incorrect behavior with regard to shared libraries and hidden symbols. There were two scenarios:

  - If a shared library referenced a symbol that existed elsewhere in both hidden and non-hidden versions, the **gold** linker produced a bogus warning message about the hidden version.

  - If a shared library referenced a symbol that existed elsewhere only as a hidden symbol, the **gold** linker created an executable, even though it could not work.

  The **gold** linker has been updated so that it no longer issues bogus warning messages about hidden symbols that also exist in a non-hidden version. The second scenario cannot be solved in the linker. It is up to the programmer to ensure that a non-hidden version of the symbol is available when the application is run.

  As a result, the two linkers' behavior is closer, but they still differ in case of a reference to a hidden symbol that cannot be found elsewhere in a non-hidden version. Unfortunately, there is not a single correct behavior for this situation, so the linkers are allowed to differ.

- The valgrind-openmpi subpackage is no longer provided with **Valgrind** in Red Hat Developer Toolset. The devtoolset-<version>-valgrind-openmpi subpackages previously caused incompatibility issues with various Red Hat Enterprise Linux minor releases and problems with rebuilding. Users are recommended to use the latest Red Hat Enterprise Linux system version of the valgrind and valgrind-openmpi packages if they need to run **Valgrind** against their programs that are built against the openmpi-devel libraries.

- The stap-server binary is no longer provided with **SystemTap** since Red Hat Developer Toolset 12. BZ#2099259

CHAPTER 3. ADDITIONAL RESOURCES

For more information about Red Hat Developer Toolset 12.0 and Red Hat Enterprise Linux, see the resources listed below.

3.1. RED HAT PRODUCT DOCUMENTATION

The **Red Hat Documentation** portal located at https://access.redhat.com/documentation/ serves as a central source of all product documentation. It is translated in 22 languages and for each product, it provides different kinds of books from release and technical notes to installation, user, and reference guides in HTML, PDF, and EPUB formats.

The following is a brief list of documents that are directly or indirectly relevant to this book:

- Red Hat Developer Toolset 12.0 User Guide – The *User Guide* for Red Hat Developer Toolset contains more information about installing and using this content set.

- Red Hat Software Collections 3.8 Packaging Guide – The *Packaging Guide* for Red Hat Software Collections explains the concept of Software Collections, documents the **scl** utility, and provides a detailed explanation of how to create a custom Software Collection or extend an existing one.

- Red Hat Software Collections 3.8 Release Notes – The *Release Notes* for Red Hat Software Collections document known problems, possible issues, and other important information available at the time of release of the content set. They also contain useful information on installing, rebuilding, migrating etc.

- Using Red Hat Software Collections Container Images – This book provides information on how to use container images based on Red Hat Software Collections. The available container images include applications, daemons, databases, as well as the Red Hat Developer Toolset container images. The images can be run on Red Hat Enterprise Linux 7 Server and Red Hat Enterprise Linux Atomic Host.

- Red Hat Enterprise Linux 7 Developer Guide – The *Developer Guide* for Red Hat Enterprise Linux 7 provides an introduction to application development tools in Red Hat Enterprise Linux 7.

- Red Hat Developer Tools documentation – The Red Hat Developer Tools provides the **Eclipse** Integrated Development Environment and the following compiler toolsets: Clang and LLVM Toolset, Rust Toolset, and Go Toolset.

3.2. RED HAT DEVELOPERS

- Red Hat Developer Program – The *Red Hat Developers* community portal.

- Overview of Red Hat Developer Toolset on Red Hat Developers – The *Red Hat Developers* portal provides a number of tutorials to get you started with developing code using different development technologies. This includes the Node.js, Perl, PHP, Python, and Ruby Software Collections.

- Red Hat Developer Blog – The *Red Hat Developer Blog* contains up-to-date information, best practices, opinion, product and program announcements as well as pointers to sample code and other resources for those who are designing and developing applications based on Red Hat technologies.

APPENDIX A. REVISION HISTORY

Revision 12.0-0	**Tue Nov 22 2022**	**Lenka Špačková**

Release of Red Hat Developer Toolset 12.0 Release Notes.