RogueWave
SOFTWARE

Klocwork 2019.2
Release Notes

Contents

≋ Release notes

These release notes cover Klocwork 2019.2.

Changes affecting migration

This section details product changes that affect how Klocwork data is migrated from a previous version. For general information about upgrading, see Upgrading from a previous version.

Licensing changes

2018 licenses are not compatible with Klocwork 2019.2. You need a new license to use the latest version of the product. Contact license@roguewave.com to obtain a new license.

Disabled checkers

If you chose to migrate your projects_root directory, verify that you have the same checker configuration as the previous release before your first integration build analysis.

What's new in Klocwork 2019.2

Here are the highlights for Klocwork 2019.2. If you're upgrading, also see the Limitations on page 14 for items that affect how you use Klocwork.

New taxonomy

We've added a new taxonomy that covers ISO/IEC TS 17961. ISO/IEC TS 17961 enumerates secure coding rules to diagnose insecure code beyond the requirements of the C language standard. For more information, see ISO/IEC TS 17961 C rules mapped to Klocwork Checkers.

Support for Visual Studio 2019

We're pleased to announce that we now support Visual Studio 2019 (Visual Studio Extension).

MISRA C 2012 new rules

Our enhanced coverage for MISRA C 2012 now includes checkers to support Rules 21.13 and 21.19.

Move to Open JDK

We've improved our support for Java by migrating to Open JDK on Linux and Windows.

Clearer build analysis reporting

Building on our changes from release 2019.1, we've extended our build analysis reporting and logging improvements to our Desktop analysis tools.

Build analysis improvements

Projects using multiple compilers will now see more accurate analysis results for C++ 14 and C++ 17. We've also improved data tracking in inter-procedural analysis when opaque pointer interfaces are used.

Changes to PATH API

We've added a new function into the kpa API: getFrontendLanguage(), as well as two new constants: LANGUAGE_C and LANGUAGE_CXX.

Updates to help

We've updated the look and feel of the help we include with the Portal and desktop analysis plug-ins.

Licensing

2018 licenses are not compatible with Klocwork 2019.2. You need a new license to use the latest version of the product. Contact license@roguewave.com to obtain a new license.

Improvements to supported compilers

We've improved support for the following compilers:

- Archelon CSR Kalimba C
- Clang
- GNU
- Green Hills
- IAR Systems C compiler/linker for ARM

For the full list of supported C/C++ compilers, see C/C++ compilers supported for build integration.

Checker improvements

From release to release, we improve issue detection to bring state-of-the-art capabilities to our customers. As a result, expect your analysis results to change as accuracy and coverage improve.

New checkers

Checker	Description
MISRA.STDLIB.CTYPE.RANGE.2012_AMD1	Implements MISRA C 2012 Rule 21.13: Any value passed to a function in `<ctype.h>` shall be representable as an *unsigned char* or be the value of `EOF`.
MISRA.STDLIB.ILLEGAL_WRITE.2012_AMD1	Implements MISRA C 2012 Rule 21.19: The pointers returned by the Standard Library functions *localeconv*, *getenv*, *setlocale* or, *strerror* shall only be used as if they have pointer to const-qualified type.

Enabled or disabled checkers

No changes were made to the default `enabled` field of the checker configuration files for this release.

Taxonomy improvements

As part of our installation, we offer several custom taxonomy files that map our checkers to standards such as MISRA, CWE, OWASP and DISA STIG.

Taxonomy	Updates
iso_iec_ts_17961_c.tconf and iso_iec_ts_17961_c_ja.tconf	These are new taxonomies that map Klocwork checkers to the ISO/IEC TS 17961 C secure coding rules.
misra_c_2012_with_amd1_c90.tconf and misra_c_2012_with_amd1_c90_ja.tconf	We added references to the following checkers: MISRA C 2012 Rule 21.13: Any value passed to a function in `<ctype.h>` shall be representable as an *unsigned char* or be the value of `EOF`. • MISRA.STDLIB.CTYPE.RANGE.2012_AMD1 MISRA C 2012 Rule 21.19: The pointers returned by the Standard Library functions *localeconv*, *getenv*, *setlocale* or, *strerror* shall only be used as if they have pointer to const-qualified type. • MISRA.STDLIB.ILLEGAL_WRITE.2012_AMD1

Taxonomy	Updates
misra_c_2012_with_amd1_c99.tconf and misra_c_2012_with_amd1_c99_ja.tconf	We added references to the following checkers:
	MISRA C 2012 Rule 21.13: Any value passed to a function in `<ctype.h>` shall be representable as an *unsigned char* or be the value of `EOF`.
	• MISRA.STDLIB.CTYPE.RANGE.2012_AMD1
	MISRA C 2012 Rule 21.19: The pointers returned by the Standard Library functions *localeconv*, *getenv*, *setlocale* or, *strerror* shall only be used as if they have pointer to const-qualified type.
	• MISRA.STDLIB.ILLEGAL_WRITE.2012_AMD1

End of support announcement

Klocwork 2018.3 was the last supported release of the Klocwork Portal and License Server on the following platforms: AIX, Mac, and Solaris. Rogue Wave is also here to help! If you need help moving your Klocwork Portal or License Server to Linux or Windows, you can contact Rogue Wave Professional Services to discuss assistance via a services engagement.

Changes to system requirements

This section lists changes to the System requirements. We've added support for the following:

- Debian 9.9
- Ubuntu 18.04.2 LTS and 19.04
- Fedora 30
- SUSE Enterprise/Leap 15
- macOS 10.14.4
- Eclipse 4.12
- Android Studio 3.4
- Visual Studio 2017 up to version 15.9.11
- Visual Studio 2019 up to version 16.0.3
- IntelliJ IDEA 2018.x up to 2018.3.6, 2019.1
- Internet Explorer 11.0.x to 11.0.115
- Mozilla Firefox 66.x.x
- Google Chrome versions 54.x to 74.x
- Safari 9.1.x to 12.1
- Jenkins 1.58 to 2.174
- TeamCity 9.13 to 2018.2.4

We now ship Open JDK 1.8.0_202 with Linux and Windows.

Klocwork now uses MariaDB 10.0.38.

Changes to commands, tools, and options

We added an option to enable/disable Code Review in the Visual Studio Extension.

We've introduced the `--short log` option for the Eclipse and Eclipse-based plugins, Klocwork Desktop, and the kwcheck and kwciagent commands.

For more information about Klocwork commands, see Command Reference.

What's new in Klocwork 2019.1

Here are the highlights for Klocwork 2019.1. If you're upgrading, also see the Limitations on page 14 for items that affect how you use Klocwork.

Improvements to security vulnerability detection

Our improved security checkers catch vulnerabilities related to the tracking of tainted data used in structures, after offsets, and when casting.

OWASP 2017 Top 10 Security Risks taxonomy

We've added a new Java taxonomy that covers the OWASP Top 10 Security Risks for 2017. For more information, see OWASP Top 10 Security Risks for 2017 mapped to Klocwork Java checkers.

Analysis engine enhancement

Our Linux platform now leverages 64-bit architecture to generate build specifications, allowing Klocwork to handle larger and more complex code bases and projects.

MISRA C 2012 new and improved

We've enhanced coverage for MISRA C 2012 by adding a checker to support Rule 18.1 and improved the checker that supports Rule 19.1.

Clearer build analysis reporting

We've made improvements to the way Klocwork reports a successful clean build by providing clearer information to users on the quality of analysis and if it requires review. We've also added a new option so that you can optimize and reduce the size of your build.log file.

Improved support for Microsoft Visual Studio

Our Visual Studio Extension now supports a broader range of Visual Studio 2017 update versions and includes general improvements to stability and performance.

Licensing

2018 licenses are not compatible with Klocwork 2019.2. You need a new license to use the latest version of the product. Contact license@roguewave.com to obtain a new license.

Improvements to supported compilers

We've improved support for the following compilers:

- ARM Optimizing C/C++ compiler (formerly TI tms470 C/C++ compiler)
- Clang
- GNU
- Green Hills
- Microsoft Visual C++
- Mono Headset SDK
- Nvidia CUDA
- Plan 9 C
- WinAVR

For the full list of supported C/C++ compilers, see C/C++ compilers supported for build integration.

Checker improvements

From release to release, we improve issue detection to bring state-of-the-art capabilities to our customers. As a result, expect your analysis results to change as accuracy and coverage improve.

New checkers

Checker	Description
MISRA.PTR.ARITH.NOT_SAME.2012	Implements MISRA C 2012 Rule 18.1: A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand

Modified checkers

Checker	Description
MISRA.ASSIGN.OVERLAP	New defects detected.
RLK.SQLOBJ	Fewer false positives are expected.
SV.TAINTED.ALLOC_SIZE	New defects detected.
SV.TAINTED.BINOP	New defects detected.
SV.TAINTED.CALL.BINOP	New defects detected.
SV.TAINTED.CALL.DEREF	New defects detected.
SV.TAINTED.CALL.INDEX_ACCESS	New defects detected.
SV.TAINTED.CALL.LOOP_BOUND	New defects detected.
SV.TAINTED.DEREF	New defects detected.
SV.TAINTED.INDEX	New defects detected.
SV.TAINTED.INDEX_ACCESS	New defects detected.

Enabled or disabled checkers

No changes were made to the default `enabled` field of the checker configuration files for this release.

Taxonomy improvements

As part of our installation, we offer several custom taxonomy files that map our checkers to standards such as MISRA, CWE, OWASP and DISA STIG.

Taxonomy	Updates
misra_c_2012_c90.tconf and misra_c_2012_c90_ja.tconf	We added references to the following checker: MISRA C 2012 Rule 18.1: A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand • MISRA.PTR.ARITH.NOT_SAME.2012
misra_c_2012_c99.tconf and misra_c_2012_c99_ja.tconf	We added references to the following checker: MISRA C 2012 Rule 18.1: A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand • MISRA.PTR.ARITH.NOT_SAME.2012
misra_c_2012_with_amd1_c90.tconf and misra_c_2012_with_amd1_c90_ja.tconf	We added references to the following checker:

Taxonomy	Updates
	MISRA C 2012 Rule 18.1: A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand • MISRA.PTR.ARITH.NOT_SAME.2012
misra_c_2012_with_amd1_c99.tconf and misra_c_2012_with_amd1_c99_ja.tconf	We added references to the following checker: Implements MISRA C 2012 Rule 18.1: A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand • MISRA.PTR.ARITH.NOT_SAME.2012
owasp_2017_10_java.tconf and owasp_2017_10_java_ja.tconf	These are new taxonomies that map Klocwork Java checkers to the OWASP 2017 Top 10 Security Risks.

End of support announcement

Klocwork 2018.3 was the last supported release of the Klocwork Portal and License Server on the following platforms: AIX, Mac, and Solaris. Rogue Wave is also here to help! If you need help moving your Klocwork Portal or License Server to Linux or Windows, you can contact Rogue Wave Professional Services to discuss assistance via a services engagement.

Changes to system requirements

This section lists changes to the System requirements. We've added support for the following:

- Debian 9.x to 9.8
- Red Hat Enterprise Linux 7.6
- Ubuntu 16.04.5 LTS and 18.10
- Windows 10 1709 to 1809
- Windows Server 2019
- IBM AIX 7.2 TL1 to TL3
- Eclipse 3.4 to 4.11
- Microsoft Visual Studio 2017 up to version 15.9.9
- Android Studio 1.0 to 3.3 (3.3.1)
- JetBrains IntelliJ IDEA 11.x to 2018.x (2018.3.3)
- Jenkins 1.58 to 2.165
- TeamCity 9.1.3 to 2018.2.3
- Google Chrome 54.x to 72.x
- Mozilla Firefox 65.x.x.
- Microsoft Edge 44.x to 44.17763
- Microsoft Internet Explorer 11.0.x to 11.0.105

Changes to commands, tools, and options

In Klocwork Desktop, we modified the description of the check box that you can select if you want to leave Klocwork Desktop running in background when the window is closed.

We modified the kwbuildproject command. We added a `--short-log` option that produces a stripped down version of the build.log file.

We also modified the `--strict` option so that when `--strict` is specified, extra warning information will be printed to the parse_errors.log and build_integration.log files. If kwbuildproject skips one or more functions during analysis, additional information will be printed to the function_information.log file.

For more information about Klocwork commands, see Command Reference.

What's new in Klocwork 2019

Here are the highlights for Klocwork 2019. If you're upgrading, also see the Limitations on page 14 for items that affect how you use Klocwork.

Analysis engine and compiler improvements

We've improved support for the Microsoft Visual C++ compiler to support modern language features and defect detection for C++ 11, C++14, and C++17 both locally, using MS Visual Studio, and remote server analysis.

Improved Microsoft Visual Studio Support

Our Visual Studio Extension now supports analysis of C++ 11, C++14, and C++17 language features so you can find new defects locally on your desktop. For more information, see Supported C++ language specifications.

Improved cross-version support for builds

Klocwork 2019 has further expanded cross-version support for builds, up to a full major release back. This means you can load Klocwork 2018.x builds into Klocwork 2019 without having to import or migrate data. For large organizations, this feature provides flexibility by allowing you to upgrade the Portal and Server to take advantage of improvements, while still analyzing some of your projects with a previous version of Klocwork. For more information, see Cross-version support for builds.

Introduction of the Build Tools package

We now offer a lightweight Klocwork Build Tools package which can be deployed on all supported platforms. For more information, see Installing the Build Tools package.

Licensing

2018 licenses are not compatible with Klocwork 2019.2. You need a new license to use the latest version of the product. Contact license@roguewave.com to obtain a new license.

Maintenance for Klocwork 2017 ending

Maintenance for all versions of Klocwork 2017 is ending: the end of maintenance (EOM) date is February 20, 2019; the end of sale (EOS) date is also February 20, 2019. For information about the availability of support for any release of Klocwork see the Rogue Wave Product Lifecycle.

Use kwinject instead of kwvcprojparser

If you have been using the kwvcprojparser command to generate a build specification, Rogue Wave strongly recommends you use the kwinject command going forward. For more information, see kwinject.

Improvements to supported compilers

We've improved support for the following compilers:

- Clang
- HiveCC
- Microsoft Visual C++
- Paradigm C/C++

For the full list of supported C/C++ compilers, see C/C++ compilers supported for build integration.

Checker improvements

From release to release, we improve issue detection to bring state-of-the-art capabilities to our customers. As a result, expect your analysis results to change as accuracy and coverage improve.

Modified checkers

Checker	Description
MISRA.FUNC.NOPROT.CALL	Fewer false positives are expected.
MISRA.FUNC.NODECL.CALL.2012	Fewer false positives are expected.

Enabled or disabled checkers

No changes were made to the default `enabled` field of the checker configuration files for this release.

Taxonomy improvements

As part of our installation, we offer several custom taxonomy files that map our checkers to standards such as MISRA, CWE, OWASP and DISA STIG. There were no changes to the taxonomy files in this release.

End of support announcement

Klocwork 2018.3 was the last supported release of the Klocwork Portal and License Server on the following platforms: AIX, Mac, and Solaris. Rogue Wave is also here to help! If you need help moving your Klocwork Portal or License Server to Linux or Windows, you can contact Rogue Wave Professional Services to discuss assistance via a services engagement.

Changes to system requirements

This section lists changes to the System requirements. Note that in some cases, we have broadened our support statement to include a range of releases (for example, 10.x, or 11.0.x), instead of specific minor releases. Supported .x versions are limited to releases available at the time of the Klocwork software release. We've added support for the following:

- Windows Server 2008 to 2016
- Ubuntu 16.04.x, 18.04.x
- Fedora 27 to 29
- OpenSUSE Tumbleweed
- SUSE Enterprise 12.x and Leap 15
- Solaris 10.x, 11.x
- macOS 10.12.x to 10.13.x
- Eclipse 3.4 to 4.10
- Android 1.0 to 3.2 (3.2.1)
- IntelliJ IDEA 11.x to 2018.x (2018.2.5)
- Internet Explorer 11.0.x
- Microsoft Edge 20.x, 25.x, 38.x, 40.x, 41.x, 42.x, 44.x
- Mozilla Firefox 60.x.x, 62.x.x
- Google Chrome 54.x to 70.x
- Apple Safari 9.1.x to 12.x
- Jenkins 1.58 to 2.148
- TeamCity 9.1.3 to 2018.1.3
- Gradle 3.x to 4.10.2
- IBM AIX 7.1 TL4 to TL5 (Server package only) and IBM AIX 7.2 TL1 to TL2 (Server package only)

We no longer support the following:

- Windows 10 Anniversary (v 1607)
- Debian 7.x (7.11)
- Red Hat Linux 6.x (6.8 to 6.10)
- CentOS 6.x (6.8 to 6.10)
- Ubuntu 14.04.5
- Fedora 25, 26
- OpenSUSE Leap 42.2, 42.3

- IntelliJ IDEA 10.0, 10.5
- Mozilla Firefox 52.7.2, 57.x, 59.0.1
- IBM AIX 7.0 (Server package only)
- macOS 10.11.6

Changes to commands, tools, and options

In the Visual Studio Extension, we added an option to change the Analysis Mode to use classic mode. When this option is enabled, it forces Klocwork to use the previous generation (pre-Klocwork 2018) analysis engine. This option matches the behavior of our other IDE plugins on Eclipse and Eclipse-based IDEs.

For more information about Klocwork commands, see Command Reference.

Fixed issues in Klocwork 2019.2

The following issues were fixed in Klocwork 2019.2.

General issues

Number	Description
00038357, SUPPORT-8716	Fixed an issue with the way the Japanese help for some commands displayed.
SUPPORT-8800, SUPPORT-9425, SUPPORT-8611, SUPPORT-8025	Improved support for the Green Hills compiler.
SUPPORT-10761	Verified that we resolved an issue related to long command lines and build logs.
SUPPORT-23150	Improved data tracking in inter-procedural analysis when opaque pointer interfaces are used.
SUPPORT-9456	Improved support for the Clang compiler.
SUPPORT-21653	Fixed an issue with unattended installations on Windows related to the installation of the JRE. Updated the documentation to specify that you must always install the JRE component.
SUPPORT-22231, SUPPORT-9258, SUPPORT-22878	Improved support for the GNU compiler.
SUPPORT-22801	Improved support for the IAR Systems C compiler/linker for ARM.
SUPPORT-8858	Fixed an issue related to a slow-down in incremental analysis.
SUPPORT-25198	Added support for Visual Studio 2019.
SUPPORT-23139	Fixed an issue with the Visual Studio Extension that prevented a project from upgrading smoothly from Visual Studio 2010 to Visual Studio 2017.
SUPPORT-20879	Migrated our Linux and Windows platforms to use Open JDK.
SUPPORT-20533	Extended our build analysis reporting and logging improvements to our Desktop analysis tools.
SUPPORT-23521	Improved how the Visual Studio Extension discovers platform tool sets.
SUPPORT-17366	Improved support for the LLVM command `llvm-strip`.
SUPPORT-23466	Added an option to enable or disable the code review tool in the Visual Studio Extension.
SUPPORT-24235	Fixed an issue with generating a build specification when using the Eclipse plug-in.

Number	Description
SUPPORT-25234	Fixed an issue with analysis related to a default object file being added to a build specification.

Documentation issues

Number	Description
SUPPORT-21130	Explained that certain status code messages in the build log can be ignored.
SUPPORT-23868	Added detail to the SV.HTTP_SPLIT checker topic to explain that the checker is designed to make sure input validation is consistent, and recommended users implement annotations and a custom KB to do so.

Fixed issues in Klocwork 2019.1

The following issues were fixed in Klocwork 2019.1.

General issues

Number	Description
00035336, SUPPORT-7012, 00038420, SUPPORT-8800, 00038796, SUPPORT-9095, SUPPORT-21748	Improved support for the Green Hills compiler.
SUPPORT-8046	Improved the dbvalidate tool.
00037836, SUPPORT-8313	Developed 64-bit support for kwinject on Linux.
00037914, SUPPORT-8374	Reduced false negatives with the checker SV.TAINTED.INDEX.
00038349, SUPPORT-8740	Improved support for the Nvidia CUDA compiler.
SUPPORT-9381	Fixed an issue with the MISRA C++ 2008 taxonomy to remove a small amount of duplication.
00039288	Fixed an issue related to null pointer exceptions not being reported in all cases with the String startsWith method.
SUPPORT-9571	Fixed an analysis issue related to the handling of lambda expressions.
SUPPORT-10541	Reduced false negatives with the checker SV.TAINTED.CALL.INDEX_ACCESS.
SUPPORT-10879	Fixed an issue related to the linking stage of analysis.
SUPPORT-9888	Improved support for the GNU compiler.
SUPPORT-8792	Reduced false negatives with several checkers related to tainted data inside nested structures.
SUPPORT-17316	Fixed an issue related to loading tables to the database.
SUPPORT-17366	Improved support for the ar filter file related to parameter files using the @ notation.
SUPPORT-20878	Generated and packaged KB files for several Windows C# open source projects.
SUPPORT-21612	Fixed an issue with analysis related to a long list of arguments.

Number	Description
SUPPORT-21569	Fixed an issue with Klocwork Desktop to improve readability of the screen when it uses high-contrast color settings.
SUPPORT-22021	Fixed an issue with Visual Studio analysis related to having spaces in the project path.
SUPPORT-21992	Updated the Java knowledge base to correct a false positive with the checker RLK.SQLOBJ.

Documentation issues

Number	Description
SUPPORT-9376	Added guidance for using regular expressions in path patterns in modules.
00039248, SUPPORT-9472	Added more information about how wildcards work when adding paths for modules.
SUPPORT-22358	Updated the code snippets for several C# checkers to correctly identify on which line Klocwork reports a defect.
SUPPORT-23531	Corrected the Supported C++ language specifications table to properly reflect our supported language specifications for Windows.

Fixed issues in Klocwork 2019

The following issues were fixed in Klocwork 2019.

General issues

Number	Description
00038220, SUPPORT-8632	Reduced false positives with the checkers MISRA.FUNC.NOPROT.CALL and MISRA.FUNC.NODECL.CALL.2012.
00037234, SUPPORT-7795	Improved support for the Paradigm compiler.
00037505, SUPPORT-8027	Fixed an issue with the Klocwork Server related to permission lookup operations.
00038688, 00038713, SUPPORT-9032, SUPPORT-9011	Fixed an analysis issue related to the checker CWARN.MEM.NONPOD.
00036212, 00036148, SUPPORT-7083	Improved support for the Microsoft Visual C++ compiler.
00039348, SUPPORT-9555	Fixed an issue with Checker Studio related to support for multi-line patterns.
00039160	Modified the way Klocwork Desktop works so that you can change the behavior of the x icon in the window and close the application instead of minimizing to the system tray.
SUPPORT-9547, SUPPORT-9757, SUPPORT-9914, SUPPORT-9919, 00039340, 00039341,	Improved support for the Clang compiler.

Number	Description
SUPPORT-20531, SUPPORT-10964, SUPPORT-10478, 00036236, 00036133	
SUPPORT-10543	Improved the way our analysis engine handles override file options.
00038553, SUPPORT-8907	Fixed an issue with importing projects related to bugtracker IDs not being imported.
00034522	Updated the knowledge base to reduce false positives with the checker NPD.FUNC.MUST.
SUPPORT-9392	Added an option in Klocwork Desktop so that you can click the X icon in the window and close the application instead of minimizing the application.
SUPPORT-9302	Fixed an analysis issue related to a missing include.
SUPPORT-10514	Added a mechanism to add a warning to the build log if the build specification contains incomplete lines.
SUPPORT-9930	Fixed an issue with kwprojcopy related to exporting very large projects.

Documentation issues

Number	Description
00039133	Added a limitation that recommends the use of regular expressions instead of a file list with the Web API.
SUPPORT-17143	Updated the system requirements to correct the versions of Linux that we support.
SUPPORT-9868	Updated the MISRA 2008 table to correctly state that a small number of rules are "no plans to support" instead of "not statically verifiable."
SUPPORT-9671, SUPPORT-9882	Fixed a website issue that resulted in text being hidden.

Limitations

This section contains limitations added in both this release and in previous releases.

Limitations for installation, upgrade and deployment

If re-installing the Klocwork plug-in for TeamCity, make sure the Project Settings do not have the 'klocwork.step.enabled' parameter

If you have installed and uninstalled the Klocwork plug-in for TeamCity previously, this parameter may have been defined. When doing a new installation, ensure this parameter has been removed.

Workaround: To remove this parameter, access your TeamCity server and go to **Administration -> <Project_name> -> Project Settings -> Configuration Parameters**. From this page, remove the 'klocwork.step.enabled' parameter.

Limitations for Checker configuration migration

Note the following limitations with checker configuration files during the upgrade process (via the import process):

- Only modifications to default checker configuration files are imported. If you had a non-default checker enabled in an earlier installation and it was renamed in a new version, you will not see the checker in new builds. You must manually re-enable the checker in the new version of Klocwork.

- If a checker that was enabled by default was renamed in the new version of Klocwork, you will not see new codes until the first system build of the new installation.

java_wrappers.conf is no longer used to edit heap size setting

A new file, java_wrappers_memory.conf is created during installation, that populates appropriate heap sizes according to your machine's memory. If you want to modify the heap size, modify this file. The previous recommendation to modify the java_wrappers.conf on Windows is deprecated, as those settings are ignored. Similarly, the previous recommendation for Linux, to modify the last two lines in the shell scripts under <klocwork_install_path>/bin, is unsafe as it may conflict with the java_wrappers_memory.conf settings.

Limitation for importing projects with existing reports

If you attempt to import a project with existing reports that use default metric names, you may see unexpected results.

Workaround: When importing a project, ensure that the reports do not use default metric names. If you encounter this error message, you can either delete and re-create the report or edit the metrics.xml file, ensuring that missing or disabled definitions are enabled.

You must have the Microsoft .NET 4.0 Framework installed in order to run Windows services

This framework is installed by default as part of Windows 8 and Windows 10. For all other versions of Windows, you must download the Microsoft .NET 4.0 Framework Installer and install the framework manually.

kwcollect fails on tables generated by new analysis engine

The behavior of kwcollect has changed with the introduction of the Klocwork 2018 analysis engine.

Workaround: If your project has been built with Klocwork 2018's new analysis engine, you must include the, '--all-sources' option on the command-line. This requirement does not apply if your project was built without Klocwork 2018's new analysis engine. To determine if your project was built with the new analysis engine, examine the output of the build process in the build.log, contained in the root of the build's output tables folder. Find the line that begins with 'Selected Engines'. Your project has been built using Klocwork 2018's new analysis engine if 'MODERN' appears between square brackets.

Limitations for Mac OS support

- On Mac, clients running Flex Net Publisher version 11.14.0.2 cannot connect to a Klocwork 2019.2 server running Flex Net Publisher 11.14.1.2. For a workaround, see kwlef error states license is not valid.
- Distributed Analysis is not supported.
- For developers, plug-in support is provided for Eclipse and IntelliJ IDEA. If your developers are not using Eclipse or IntelliJ IDEA, they need to use Klocwork Desktop Command Line for C/C++ or Java (kwcheck) or Klocwork Desktop to analyze their code and view detected issues. See Fixing issues before check-in with Klocwork Desktop Analysis.
- System Integrity Protection (SIP) blocks the kwinject command from running properly on Mac OS X 10.10 and later. Kwinject returns the following warning, with error code 1: "`System Integrity Protection is enabled. kwinject cannot inject to process.`" *Workaround:* Disable SIP on the machine running the Klocwork analysis or see Using kwwrap plus kwinject to generate a build specification.

Limitations for build integration

Cannot load Android 4.4 (KitKat) using the default memory settings for kwloaddb, kwadmin and kwjava

When building the Android platform, you may need to increase the memory settings for certain Klocwork tools on the machine invoking the load process. These values can be modified in the `<klocwork_install>/config/java_wrappers_memory.conf` file.

Android N Java analysis with Jack toolchain

When building Android N using the Jack compiler, some jar files required for Klocwork Java analysis are not generated during the build process. Therefore, `kwbuildproject` encounters "Unresolved import", "Unresolved method", and "Unresolved name" semantic errors that affect the accuracy of the analysis results.

Workaround: Open a ticket with Klocwork customer support. Customer support can provide a script that can generate the jar files required for analysis. Run the script after running the `kwinject` command and before running the `kwbuildproject` command.

Limitations for C# analysis

Klocwork's C# analysis is supported only on Windows.

The following features are not supported for C# integration projects:

Feature	Details
Build integration	• kwinject cannot be used to create a build specification for a C# project. Instead, use kwcsprojparser. • Build specification templates
Integration build analysis	• Mixed-language projects (you need to create one C/C++ project and one C# project) • Parallel analysis • Incremental analysis
Klocwork Static Code Analysis	• "Show implementation", "Show declaration", and Source Cross-Reference
Distributed analysis	• Distributed analysis is not supported for C#.

The following features are not supported for C# desktop analysis:

- On-the-fly analysis
- Display of server issues in Visual Studio
- Parallel analysis
- Incremental analysis
- File-level analysis in Visual Studio (only solutions and projects can be analyzed)
- Using metric thresholds and knowledge bases

Using metric thresholds and knowledge bases is not supported for C# server build analysis.

Limitations for Klocwork Static Code Analysis

Left-navigation not clickable in offline help when using Internet Explorer

If you are using Internet Explorer, the items in the left navigation pane of the Portal offline help are not clickable.

Workaround: Use search to locate topics of interest, or use any of the other supported browsers.

In Microsoft Edge, some items may not be clickable

Due to a Microsoft Edge issue, some items in the portal may not be clickable. For more information, see https://developer.microsoft.com/en-us/microsoft-edge/platform/issues/5782378/

Workaround: Refresh the page.

Limitations for Klocwork Desktop Analysis

Analysis is not supported for 'no-resolve' mode in certain scenarios

The "no-resolve" mode was added to support symbolic links to source files on Linux. Symbolic links to directories are not supported.

The Eclipse plug-in supports the "no-resolve" mode only if project is configured to use an external build specification, and that build specification was created by using kwinject with "--no-resolve" option.

For WindRiver Workbench users, you will receive an error message if you attempt to use a project with exterior sources linked to it.

Limitations for the Visual Studio plug-in

Visual Studio hang

The Klocwork development team is tracking a support request with the Visual Studio Technical Support team where user actions cause Visual Studio to hang under a number of conditions. These Visual Studio hangs occur whether or not the Klocwork VS Extension is installed. For example, when navigating into the definition of a function that is defined in a source file that is not currently open in a tab in Visual Studio, Visual Studio opens that file in a temporary tab. When this temporary tab is open, if you then navigate to the definition of a different function, Visual Studio hangs.

Visual Studio 2017 help

As of Visual Studio 2017, the Help Viewer component is no longer installed by default and must be explicitly selected during installation. If you attempt to install our Klocwork extension for Visual Studio and you do not have this component installed, you will receive an error as our local help is unable to be installed. For more details on this, see Klocwork Help registration could not acquire the location of the Help Viewer.

Kwvcprojparser not supported for Visual Studio 2017

The kwvcprojparser command is not supported for Visual Studio 2017 projects built from the command line. The kwvcprojparser command is supported on Visual Studio 2010, 2012, and 2015; however, it only provides results based on the previous generation (pre-Klocwork 2018) analysis engine.

Workaround: Use the `kwinject` command to create the build specification.

The filter by severity option in the Microsoft Visual Studio extension may not display custom severities for C++ projects

For C++ projects where you have defined custom severities, the severity filter list may not display the correct items. The list may display default severity names, or in the case where you have a mixed C++ and C# project, the list will display the C# severities. You can still use the filter, but the severity names displayed in the issue tree may not match the items you selected in the list (as the filter is applied by severity number).

After uninstalling the Klocwork Microsoft Visual Studio extension, the Klocwork help content is not removed

Due to a limitation of the Microsoft VSIX installer, Klocwork help is not removed after uninstalling the plug-in.

Workaround: You can uninstall the help files manually. Go to **Help > Add and Remove Help Content**; In the **Klocwork Inc.** section, click the **Remove** action next to **Klocwork Desktop Plugin**. You can install a future version of the plug-in without issue.

For the Microsoft Visual Studio extension, minor performance degradation when working with server issues if connection to server is lost

A lost server connection causes a delay of up to three seconds when working with server issues, for example, when opening or citing a server issue.

Workaround: Work with local issues only by clicking the "Show local issues only" button.

F1 help does not work when you attempt to open help for an issue from the Klocwork Issues window in Visual Studio for the Klocwork extension for Visual Studio

If you click on an issue in the Klocwork Issues window and attempt to open the help for it by pressing F1, the shortcut opens the incorrect help in the Help Viewer.

Workaround: Open the help for the checker by right-clicking on the issue and select **View Checker Documentation** from the **Manage <checker name> Checker** menu.

Klocwork server option fails to retrieve projects when you use a hard-coded IP address

If you use a hard-coded IP address in the Klocwork server dialog under the Klocwork options menu, the Klocwork extension for Visual Studio fails to retrieve the list of projects.

Workaround: Use the host name instead of the IP address; if this is not an option, you can add an entry in the hosts file for the IP address.

Klocwork plug-in for Android Studio installs to unexpected location

If IntelliJ IDEA 2017 and Android Studio are both installed, and you install the Android Studio plug-in, the IntelliJ IDEA path will be auto-filled instead of the Android Studio path. Klocwork automatically detects your IntelliJ IDEA directory and installs the plug-in to that location.

Workaround: If more than one installation directory is detected, you must browse to the preferred location for Android Studio manually.

Limitations for Help in Android Studio and IntelliJ IDEA 2017+

Offline help is not available for Android Studio and IntelliJ IDEA versions 2017 and newer. Offline help is available by using the Klocwork Portal. Online help is also available.

Limitations for Klocwork Desktop

Analysis is not supported with any of the following configurations:

- When a project with symbolic links is configured with an external build specification that does not have the attribute "no-resolve". If a project uses symbolic links, the user must configure the project using an external build specification, and the external build specification must be created with the "no-resolve" option passed to kwinject.
- When a project with symbolic links is configured to use the Eclipse CDT toolchain. The Eclipse plug-in does not allow the user to set a "no-resolve" option.
- When a project contains a symbolic link to a directory. The plug-in supports symbolic links to files only.

Limitations for Klocwork extensibility

C/C++ Path checker compilation makefile compatibility

The makefile generated by kwcreatechecker on Unix systems requires GNU make to build the checker. The default make installed on non-GNU systems such as AIX or Solaris may not compile Klocwork extensions for C/C++. On Windows, the makefile generated by kwcreatechecker requires nmake to build the checker.

Workaround: None.

RogueWave
SOFTWARE

Toll-free: 1.800.487.3217
Direct: 1.613.836.8899
sales@klocwork.com
support@klocwork.com

1315 West Century Drive, Suite 150, Louisville, CO 80027

www.roguewave.com

www.klocwork.com