

A decorative graphic on the left side of the page, consisting of several white, curved, overlapping shapes that resemble stylized waves or a hand reaching out, set against a blue background.

Klocwork 2018.3

Release Notes

Contents

Release notes.....	3
What's new in Klocwork 2018.3.....	3
What's new in Klocwork 2018.2.....	5
What's new in Klocwork 2018.1.....	6
What's new in Klocwork 2018.....	9
Fixed issues in Klocwork 2018.3.....	11
Fixed issues in Klocwork 2018.2.....	12
Fixed issues in Klocwork 2018.1.....	13
Fixed issues in Klocwork 2018.....	14
Limitations.....	15

Release notes

These release notes cover Klocwork 2018.3.

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

2017 licenses are not compatible with Klocwork 2018.3. 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 2018.3

Here are the highlights for Klocwork 2018.3. If you're upgrading, also see the [Limitations](#) on page 15 for items that affect how you use Klocwork.

Analysis engine improvements

We continue to enhance and improve our analysis of C++ 11, C++14, and C++17 language features on Windows and Linux platforms, improving analysis accuracy and developer productivity. For more information, see [Supported C++ language specifications](#).

Expanded support for AUTOSAR C++14

We've updated and expanded support for AUTOSAR C++14 by adding a new taxonomy. For more information, see [AUTOSAR 18-03 Standard mapped to Klocwork C/C++ checkers](#).

Licensing

2017 licenses are not compatible with Klocwork 2018.3. 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:

- Clang versions 3.6, 4.0, 5.x and 6.x
- GNU (GCC) versions 4.6, 5.4, 6.x, 7.x, and 8.x.
- Hexagon Tools
- Intel C++
- Microchip MPLAB pic24
- Microchip MPLAB pic32
- Microsoft Visual C++
- Synopsys ARC MetaWare compiler
- Tensilica Xtensa 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.NODECL.CALL.2012	Fewer false positives are expected.
MISRA.FUNC.NOPROT.CALL	Fewer false positives are expected.
MISRA.FUNC.NOPROT.DEF	Fewer false positives are expected.
MISRA.FUNC.NOPROT.DEF.2012	Fewer false positives are expected.
MISRA.SWITCH.WELL_FORMED.DEFAULT.FIRST_OR_LAST.2012	New defects detected.

Enabled or disabled checkers

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

QDP improvements

We updated our QDP package to reflect improvements in MISRA and other checkers.

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. The following is the list of changes to these files in this release:

Note: If you've imported a custom taxonomy (for example, MISRA) in a previous release, you need to import the new taxonomy file to pick up these changes.

Taxonomy file	Changes in this release
autosar_cpp.tconf and autosar_cpp_ja.tconf are now named autosar_cpp_17_10.tconf and autosar_cpp_17_10_ja.tconf	Use this renamed taxonomy to ensure compliance with the AUTOSAR C++14 Standard, release 17-10.
autosar_cpp_18_03.tconf and autosar_cpp_18_03_ja.tconf	These are new taxonomies that you can use to ensure compliance with the AUTOSAR C++14 Standard, release 18-03.

End of support announcement

Klocwork 2018.3 is the last supported release of the Klocwork Portal and License Server on the following platforms: AIX, Mac, and Solaris. If you are using these platforms, we recommend you begin planning for this change now.

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 9.5
- Red Hat Enterprise Linux 6.10
- OpenSUSE Leap 15
- SUSE Linux Enterprise 15
- CentOS 6.10
- macOS 10.13.6
- Eclipse 4.8
- Android Studio 3.1 (3.1.3)
- Visual Studio 2017 update 15.8.4
- IntelliJ IDEA 2016.3.8, 2017.1.6, 2017.2.7, 2017.3.5, 2018.1.6, 2018.2
- Microsoft Internet Explorer 11.0.75
- Mozilla Firefox 52.9.0esr, 60.1.0esr, 61.0.1

- Google Chrome 68.0.3359
- Apple Safari 11.1.2
- Jenkins 2.135
- TeamCity 2018.1.1
- glibc 2.28
- gradle 4.9

We've updated the version of Apache Tomcat to 7.0.90 for Linux, Mac, and Windows.

We updated the version of Jython to version 2.7.1. This version of Jython allows customers to use some Python features in review_action.py. If you have implemented third-party bug tracking scripts and libraries, you may need to update the scripts to take advantage of these features.

Changes to commands, tools, and options

We have not modified any commands or tools in this release.

For more information about Klocwork commands, see [Command Reference](#).

What's new in Klocwork 2018.2

Here are the highlights for Klocwork 2018.2. If you're upgrading, also see the [Limitations](#) on page 15 for items that affect how you use Klocwork.

Analysis engine improvements

We're pleased to announce that our new analysis engine now provides full support for C++ 11, C++14, and C++17 language features on Windows platforms running Klocwork Server or desktop analysis tools. For more information, see [Supported C++ language specifications](#).

Licensing

2017 licenses are not compatible with Klocwork 2018.2. 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:

- Clang
- Fujitsu FR Family
- GNU (GCC)
- IAR compiler/linker for MAXQ
- QNX
- Wind River Diab

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.VAR.NEEDS.CONST	Fewer false positives are expected.
NPE.CONST	Fewer false positives are expected.
NPD.GEN.MUST	More defects detected.
SV.SERIAL.INON	More defects detected.

Checker	Description
UNINIT.STACK.MIGHT	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 taxonomy files in this release.

Changes to system requirements

This section lists changes to the [System requirements](#). We've added support for the following:

- Windows 10 version 1803
- Red Hat Enterprise Linux 7.5
- Ubuntu 16.04.4 and 18.04
- Fedora 28
- CentOS 7.5 (1804)
- macOS 10.12.6 and 10.13.5
- Android Studio 3.1 (3.1.2)
- IntelliJ IDEA 2018.1.3
- Microsoft Internet Explorer 11.0.60
- Microsoft Edge 42.17134
- Firefox 52.8.0esr, 60.0.1esr
- Google Chrome 66.0.3359
- Apple Safari 11.1
- Jenkins 2.122
- TeamCity 2017.2.4
- glibc 2.27
- gradle 4.7
- MySQL for all platforms except AIX is now 5.6.40
- The JVM bundled with Windows, Linux, and AIX is now Java 8 Update 171.

Changes to commands, tools, and options

We added an option in the Eclipse plug-in and in Klocwork Desktop that can force Klocwork to use the previous generation analysis engine (pre-Klocwork 2018). The previous analysis engine provides partial support for C++11 and C++14.

For more information about Klocwork commands, see [Command Reference](#).

What's new in Klocwork 2018.1

Here are the highlights for Klocwork 2018.1. If you're upgrading, also see the [Limitations](#) on page 15 for items that affect how you use Klocwork.

AUTOSAR C++14 taxonomy

In this release, we've introduced an AUTOSAR C++14 taxonomy that you can use to ensure compliance with the AUTOSAR C++14 Standard, release 17-10. For more information, see [AUTOSAR Standard mapped to Klocwork C/C++ checkers](#).

Analysis engine improvements

Our commitment to supporting the latest C++ standards continues. In this release we're thrilled to announce that we have full support for all C++17 language features. For more information, see [Supported C++ language specifications](#).

Web API metrics

We've added an optional parameter to the Web API metrics request called `exclude_system_files`. When set to true, the request omits system files from the metrics. For more information, see [Issue and metric API examples](#).

Licensing

2017 licenses are not compatible with Klocwork 2018.1. 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 added support for the following compilers:

- IAR compiler/linker for STM8 Microcontroller family

We've improved support for the following compilers:

- Clang
- GNU (GCC)

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.ILLEGAL_REUSE.2012_AMD1	Implements MISRA C 2012 Rule 21.20: The pointer returned by the Standard Library functions <i>asctime</i> , <i>ctime</i> , <i>gmtime</i> , <i>localtime</i> , <i>localeconv</i> , <i>getenv</i> , <i>setlocale</i> or <i>strerror</i> shall not be used following a subsequent call to the same function.

Modified checkers

Checker	Description
MISRA.CONV.NUM.NARROWER	More defects detected.
MISRA.CVALUE.IMPL.CAST.CPP	Fewer false positives are expected.
MISRA.FOR.INCR.CHANGE	Fewer false positives are expected.
MISRA.IF.NO_ELSE	Fewer false positives are expected.
MISRA.SWITCH.WELL_FORMED.DEFAULT.FIRST_OR_LAST.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. The following is the list of changes to these files in this release:

Note: If you've imported a custom taxonomy (for example, MISRA) in a previous release, you need to import the new taxonomy file to pick up these changes.

Taxonomy file	Changes in this release
autosar_cpp.tconf and autosar_cpp_ja.tconf	These are new taxonomies that you can use to ensure compliance with the AUTOSAR C++14 Standard, release 17-10.
disa_stig_v4_java.tconf and disa_stig_v4_java_ja.tconf	We added references to the following checkers: APSC-DV-002030: Encryption <ul style="list-style-type: none"> • SV.HASH.NO_SALT • SV.SENSITIVE.DATA • SV.SENSITIVE.OBJ • SV.WEAK.CRYPT
cwe_10_cxx.tconf and cwe_10_cxx_ja.tconf	We added references to the following checkers: CWE-1037: Processor Optimization Removal or Modification of Security-critical Code <ul style="list-style-type: none"> • SPECTRE.VARIANT1
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 21.20: The pointer returned by the Standard Library functions <i>asctime</i> , <i>ctime</i> , <i>gmtime</i> , <i>localtime</i> , <i>localeconv</i> , <i>getenv</i> , <i>setlocale</i> or <i>strerror</i> shall not be used following a subsequent call to the same function <ul style="list-style-type: none"> • MISRA.STDLIB.ILLEGAL_REUSE.2012_AMD1
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 21.20: The pointer returned by the Standard Library functions <i>asctime</i> , <i>ctime</i> , <i>gmtime</i> , <i>localtime</i> , <i>localeconv</i> , <i>getenv</i> , <i>setlocale</i> or <i>strerror</i> shall not be used following a subsequent call to the same function <ul style="list-style-type: none"> • MISRA.STDLIB.ILLEGAL_REUSE.2012_AMD1

Changes to system requirements

This section lists changes to the [System requirements](#). We've added support for the following:

- Debian 8.10 and 9.4
- OpenSUSE 12.3 (Ent)
- Ubuntu 14.04.05 and 16.04.03
- Windows 7 Service Pack 1
- Windows 10 version 1709
- IBM AIX 7.1 TL 5 and 7.2 TL 2
- Google Chrome 65.0.3325
- Mozilla Firefox 52.7.2 and 59.0.1
- Apple Safari 11.0.3
- Edge 20.1024 and 40.15063
- Internet Explorer 11.0.52
- Eclipse 4.7.3
- Android Studio 3.0.1
- JetBrains IntelliJ IDEA 2017.3.4
- QNX Software Dev Platform 7.0

Changes to commands, tools, and options

We added a new command called [kwciagent](#).

We removed the `kwscope` tool.

For more information about Klocwork commands, see [Command Reference](#).

What's new in Klocwork 2018

Here are the highlights for Klocwork 2018. If you're upgrading, also see the [Limitations](#) on page 15 for items that affect how you use Klocwork.

Features in the latest release of Klocwork 2018

In the latest release of Klocwork 2018, we're happy to announce the release of a new Klocwork checker, [SPECTRE.VARIANT1](#), that detects potential occurrences of Spectre variant 1 (CVE-2017-5753) in your code. For a detailed explanation about the checker and how it works, see [our video](#).

We've also improved support for Chromium-based browsers.

New analysis engine with support for latest C++ 17 language features

Our new analysis engine provides improved support for C++ 11, 14, and 17 language features. Improved support means you can be confident that Klocwork 2018 is performing the most complete analysis on the most complex C++ applications. For more information, see [Supported C++ language specifications](#).

Cross-version support for builds

Klocwork 2018 has decoupled the Build Server version from the Portal and Desktop tools, up to three minor releases back. This means you can load Klocwork 2017.1, 2017.2, and 2017.3 builds into Klocwork 2018 without having to import or migrate data. For large organizations, this feature provides flexibility by allowing you to upgrade the Portal and Desktop tools to take advantage of improvements, while still analyzing some or all of your projects with a previous version of Klocwork. For more information, see [Cross-version support for builds](#).

Licensing

2017 licenses are not compatible with Klocwork 2018. You need a new license to use the latest version of the product. Contact license@roguewave.com to obtain a new license.

In release 2017.3, we upgraded the version of FlexNet Publisher that we support for Windows, Linux, and Mac platforms to version 2016 R2 (11.14.1.2). The versions of FlexNet Publisher used with AIX and Sun Solaris are unchanged.

If you are using your own FlexNet Publisher license server, the Windows, Linux and Mac installations of Klocwork 2018.3 are compatible with FlexNet Publisher 2016 R2 (11.14.1.2) and later. The versions of FlexNet Publisher used by Solaris and AIX are not compatible; therefore, for example, a Klocwork integration build analysis on a Windows machine will not be able to check out a license from a license server running on Solaris or AIX.

For more information, see [Supported versions of Flex Net Publisher](#).

Improvements to supported compilers

We've improved support for the following compilers:

- Clang
- Microsoft Visual C++
- Wind River GCC

For the full list of supported C/C++ compilers, see [C/C++ compilers supported for build integration](#).

Changes to the Path API

In Klocwork 2016, we made a number of changes to the C++ version of our Path API. Chapter 2 of the Klocwork C/C++ Path Analysis API Reference contains a list of deprecated functions and provides a proposed replacement for each. As of Klocwork 2017.1, the use of deprecated functions causes compiler errors instead of compiler warnings.

If you're using deprecated functions, we recommend you migrate to supported functions now. For more information, see [Important changes to the Path API in version 11.2](#).

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.

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. The following is the list of changes to these files in this release:

Note: If you've imported a custom taxonomy (for example, MISRA) in a previous release, you need to import the new taxonomy file to pick up these changes.

Taxonomy file	Changes in this release
<code>disa_stig_10_cxx.tconf</code> and <code>disa_stig_10_cxx_ja.tconf</code>	We removed references to the following checkers: APP3570: Command injection <ul style="list-style-type: none">SV.EMAILSV.EXECSV.EXEC.DIRSV.EXEC.ENV APP3590.2: Buffer overflows <ul style="list-style-type: none">SV.TAINT_NATIVE
<code>disa_stig_10_java.tconf</code> and <code>disa_stig_10_java_ja.tconf</code>	We added references to the following checkers: APP3570: Command injection vulnerabilities <ul style="list-style-type: none">SV.EMAILSV.EXECSV.EXEC.DIRSV.EXEC.ENV We removed references to the following checkers: APP3760 and APP3780: Application level DoS <ul style="list-style-type: none">SV.EXECSV.EXEC.ENV

Changes to system requirements

This section lists changes to the [System requirements](#). We've added support for the following:

- Debian 9.2
- Fedora 27
- CentOS 6.9
- macOS High Sierra 10.13
- Android Studio 3.0
- IntelliJ IDEA 2017.2.6
- Internet Explorer 11.0.47
- Edge 41.16299.15
- Mozilla Firefox 57
- Google Chrome 62.0.3202

We've removed support for the following:

- Debian 7.9, 8.5, 9.1
- Red Hat Enterprise Linux 5.11, 6.8
- Ubuntu 16.10, 17.04
- Fedora 23, 24
- OpenSUSE Enterprise 11.2, 11.4
- CentOS 6.7
- macOS 10.10.5
- Visual Studio 2008
- Internet Explorer 11.0.9600, 11.0.10240
- Edge 40.15063
- Mozilla Firefox 55.0.3
- Google Chrome 61.0.3163
- Glibc below version 2.15

Changes to commands and options

We modified the `kwbuildproject` command by removing the `--log-file` and `--resume` options.

We modified the `kwbuildproject` command by adding the `--classic` option. The `--classic` option forces Klocwork to use the previous generation (pre-Klocwork 2018) analysis engine. The previous analysis engine only provides partial support for C++11 and C++14.

For more information about Klocwork commands, see [Command Reference](#).

Fixed issues in Klocwork 2018.3

The following issues were fixed in Klocwork 2018.3.

General issues

Number	Description
00038583	Fixed an issue related to the <code>kwunpack.py</code> script so that it writes build spec lines individually instead of concatenating them.
00037568	Fixed an issue with the Visual Studio 2017 Extension related to includes.
00038263	Fixed an issue with the Java compiler related to resolving non-standard compiler paths.
00038510	Fixed an issue with the Category Details report related to the LOC count being calculated incorrectly.
00035231, 00036103, 00036498, 00036712, 00036809, 00037044, 00037306, 00037630, 00038221, 00038298, 00038386, 00038481, 00038499, 00038717, 00038704, 00038711, 00038768	Improved support for the GNU (GCC) compiler.

Number	Description
00038440, 00039186	Improved support for the Clang compiler.
00038730	We updated the version of Apache Tomcat to 7.0.90 for Linux, Mac, and Windows.
00038988, 00039154, 00039155	We updated our QDP package to fix an issue related to MISRA tests.
00038924	Fixed an issue with the KAST built-in function <code>hasCompleteDefinition()</code> .
00037626	Fixed an issue with the <code>kwcheck</code> command on the Mac platform.
00037888	Fixed an issue with the Entities menu where it was not displaying all options for some projects.
00038571, 00038208	Fixed an issue with the Visual Studio Extension related to system headers.
00038639	Reduced false positives with the following checkers: <ul style="list-style-type: none"> • MISRA.FUNC.NODECL.CALL.2012 • MISRA.FUNC.NOPROT.CALL • MISRA.FUNC.NOPROT.DEF • MISRA.FUNC.NOPROT.DEF.2012
00039111	Fixed an issue with <code>kwbuildproject</code> on Windows related to UNC paths.

Documentation issues

Number	Description
00038856	Improved the description of how Klocwork calculates the NOINDPATHS metric.
00038957	Updated a link to documentation on the Developer Network website.

Fixed issues in Klocwork 2018.2

The following issues were fixed in Klocwork 2018.2.

General issues

Number	Description
00035311	Reduced false positives for the checker <code>NPE.CONST</code> .
00037739	Fixed an issue with the <code>kwinject</code> command where <code>.NET</code> config files were not loading properly.
00038399, 00038189	Updated the <code>dbvalidate</code> tool to identify and fix <code>END_TIME=null</code> issues.
00038198	Improved support for the Fujitsu FR Family compiler by adding support for <code>fcc896s</code> .
00037362, 00035713, 00035163, 00036824	Provided a workaround to resolve a FlexLM issue.
00032896, 00036409	Improved support for the Windriver Diab compiler.
00038114	Updated a knowledge base file shipped with Klocwork to include additional entries for ABV issue detection.
00037897, 00037724	Fixed an analysis issue related to a small number of MISRA checkers.
00038240	Fixed a build issue related to how the <code>kwgather</code> command handled a duplicate entry.
00037545	Upgraded the version of JVM shipped with Windows, Linux, and AIX.

Number	Description
00038054	Fixed an issue with the kwxsync command related to extra long nested comments.
00036466	Reduced false positives with the checker MISRA.VAR.NEEDS.CONST.
00037524	Improved support for the QNX compiler.
00038087	Fixed an issue related to generating a build spec on Windows.
00036317	Fixed an analysis issue related to compile-time dependencies.
00038361	Fixed an issue with the Visual Studio Extension related to C# analysis.
00038482, 00038554	Upgraded the version of MySQL for Window, Linux, Mac, and Solaris to 5.6.40.
00037127	Reduced false negatives for the checker NPD.GEN.MUST.
00035924	Fixed an issue with the Klocwork Portal related to the false positive report.
00038677	Fixed an issue for some GNU projects where "missing include files" were not actually missing.

Documentation issues

Number	Description
00038344	Fixed a trailing space that caused a broken cross-reference.

Fixed issues in Klocwork 2018.1

The following issues were fixed in Klocwork 2018.

General issues

Number	Description
00035763	Fixed an issue with the dbvalidate script related to the "owner changed by system" record overriding the status copied by the xsync command.
00035985	Fixed a build issue related to long package names.
00035089	Improved support for Incredibuild 8.01.
00034776	Fixed an issue with XML output related to output encoding.
00034795	Added text to the Linux and Solaris installers to tell customers to check for missing software dependencies.
00035416	Fixed an issue with duplicate projects related to kwmatch that prevented the project from being added properly to an issue matching group.
00034761	Removed the --log-file option from the kwbuildproject command.
00034903	Improved kwbuildproject logging so that the command logs a warning if the number of defects for an individual checker exceeds 5000.
00036569	Improved support for the Wind River GCC compiler.
00036495	Fixed an issue with the Visual Studio plug-in related to closing connected solutions.
00036308	Fixed an issue with kwcc related to having the command use kwpython instead of a version of python installed on the system.
00036566	Fixed an issue with the Eclipse plug-in.
00036935	Fixed an issue where a few Java checkers were listed in the DISA STIG version 3 taxonomy.

Number	Description
00036302	Fixed an issue with the Visual Studio plug-in related to the analysis of C++ source code in Visual Studio 2013.
00035964	Fixed an issue with the Visual Studio plug-in related to the analysis of mixed C++/C# projected in Visual Studio 2012.
00036300	Fixed an issue with the Visual Studio plug-in related to the analysis issue related to the analysis of C# projects in Visual Studio 2015.
00035334	Updated the Windows build packaging to include a missing taxonomy.
00036286	Fixed an issue with the VS plug-in related to the installing on Windows 10 in administrator mode.

Documentation issues

Number	Description
00036301	Clarified which ports are used by the Klocwork servers.
00036537	Improved the instructions for deploying custom checkers to the build machines as well as the Klocwork Server.
00036740	Corrected an error in the MISRA-C++ 2008 reference pages that caused a small number of rules to be missing.
00036448	Removed the reference to the checker MISRA.BITFIELD.SIGNED.UNNAMED from MISRA-C++ Rule 9-6-4.
00036985	Fixed a couple of broken cross-references in the online documentation.
00036640	Corrected the fixed code example for the checker SV.STRBO.BOUND_COPY.UNTERM.
00036997	Added the topic about the Klocwork Properties dialog box to the online Help.

Fixed issues in Klocwork 2018

The following issues were fixed in Klocwork 2018.

General issues

Number	Description
00035763	Fixed an issue with the dbvalidate script related to the "owner changed by system" record overriding the status copied by the xsync command.
00035985	Fixed a build issue related to long package names.
00035089	Improved support for Incredibuild 8.01.
00034776	Fixed an issue with XML output related to output encoding.
00034795	Added text to the Linux and Solaris installers to tell customers to check for missing software dependencies.
00035416	Fixed an issue with duplicate projects related to kwmatch that prevented the project from being added properly to an issue matching group.
00034761	Removed the --log-file option from the kwbuildproject command.
00034903	Improved kwbuildproject logging so that the command logs a warning if the number of defects for an individual checker exceeds 5000.
00036569	Improved support for the Wind River GCC compiler.

Number	Description
00036495	Fixed an issue with the Visual Studio plug-in related to closing connected solutions.
00036308	Fixed an issue with kwcc related to having the command use kwpython instead of a version of python installed on the system.
00036566	Fixed an issue with the Eclipse plug-in.
00036935	Fixed an issue where a few Java checkers were listed in the DISA STIG version 3 taxonomy.
00036302	Fixed an issue with the Visual Studio plug-in related to the analysis of C++ source code in Visual Studio 2013.
00035964	Fixed an issue with the Visual Studio plug-in related to the analysis of mixed C++/C# projected in Visual Studio 2012.
00036300	Fixed an issue with the Visual Studio plug-in related to the analysis issue related to the analysis of C# projects in Visual Studio 2015.
00035334	Updated the Windows build packaging to include a missing taxonomy.
00036286	Fixed an issue with the VS plug-in related to the installing on Windows 10 in administrator mode.

Documentation issues

Number	Description
00036301	Clarified which ports are used by the Klocwork servers.
00036537	Improved the instructions for deploying custom checkers to the build machines as well as the Klocwork Server.
00036740	Corrected an error in the MISRA-C++ 2008 reference pages that caused a small number of rules to be missing.
00036448	Removed the reference to the checker MISRA.BITFIELD.SIGNED.UNNAMED from MISRA-C++ Rule 9-6-4.
00036985	Fixed a couple of broken cross-references in the online documentation.
00036640	Corrected the fixed code example for the checker SV.STRBO.BOUND_COPY.UNTERM.
00036997	Added the topic about the Klocwork Properties dialog box to the online Help.

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 2017.3 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 kwrap 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	<ul style="list-style-type: none">kwinject cannot be used to create a build specification for a C# project. Instead, use kwcsprojparser.Build specification templates
Integration build analysis	<ul style="list-style-type: none">Mixed-language projects (you need to create one C/C++ project and one C# project)Parallel analysisIncremental analysis
Klocwork Static Code Analysis	<ul style="list-style-type: none">"Show implementation", "Show declaration", and Source Cross-Reference
Distributed analysis	<ul style="list-style-type: none">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

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.

Middle-clicking a link doesn't open it in a new tab when using Google Chrome

Due to a bug in Google Chrome, some links do not open in a new tab when they are middle clicked, shift-clicked or ctrl-clicked after the first time the link is opened in this manner. Each successive attempt simply opens the link within the active tab. For more information, see <http://code.google.com/p/chromium/issues/detail?id=177502>.

Workaround: Refresh the page and this will allow you to open the link in a new tab the first time you attempt it.

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 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.

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 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.



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



This document, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information contained herein is the property of Rogue Wave Software, Inc. and is confidential between Rogue Wave Software, Inc. and the client and remains the exclusive property of Rogue Wave Software, Inc. No part of this documentation may be copied, translated, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Rogue Wave Software, Inc. If you find any problems in the documentation, please report them to us in writing. Rogue Wave Software, Inc. does not warrant that this document is error-free.

Klocwork is a registered trademark of Rogue Wave Software, Inc.

All other trademarks are the property of their respective owners. All help content for Klocwork's MISRA checkers is copyright by MIRA Ltd, on behalf of the MISRA Consortium.

Copyright notices for third-party software are contained in the file THIRDPARTYLICENSEREADME.txt, located in the Klocwork installation directory.