



Klocwork 2023.4 Release Notes

Document Number: KW2023_4 _009

Certified version: Klocwork 2023.4

Author	Revision	Comments	Date
MTooke	0.1	Initial draft created	May-30-13
MTooke	0.2	Added table describing MISRA archives	July-02-13
AWeekes	0.3	Updated for 2015 re-certification	May-29-15
SBommaganti	1.0	Updated for Klocwork 2016	Mar-3-16
SBommaganti	1.1	Updated fixed issues to include content from latest Klocwork 2016 build (11.0.1)	Mar-22-16
SBommaganti	1.2	Updated for Klocwork 2016.1	June-25-16
SBommaganti	1.3	Updated for Klocwork 2016.3	Nov-02-16

SBommaganti	1.4	Added Fixed Issues for Klocwork 2016.3	Jan-18-17
SBommaganti	2.0	Updated for Klocwork 2017	Mar-2-17
SBommaganti	2.1	Updated for Klocwork 2017.1	July-5-17
SBommaganti	2.2	Fixed a typo – reference to Klocwork 2017.2 in Klocwork 2017.1 document	Sep-7-17
SBommaganti	2.3	Updated for Klocwork 2017.2	Oct-16-17
SBommaganti	2.4	Updated for Klocwork 2017.3	Nov-2-17
MTofinetti	3.0	Updated for Klocwork 2018	2018-05-08
MTofinetti	3.1	Updated for Klocwork 2018.1	2018-07-05
MTofinetti	3.2	Updated for Klocwork 2018.2	2018-09-27
MTofinetti	3.3	Updated for Klocwork 2018.3	2018-12-06
MTofinetti	4.0	Updated for Klocwork 2019	2019-03-22
MTofinetti	4.1	Updated for Klocwork 2019.1	2019-07-04
MTofinetti	4.2	Updated for Klocwork 2019.2	2019-07-30
MTofinetti	4.3	Updated for Klocwork 2019.3; IEC 62304 added	2019-12-12
LRobertson	5.0	Updated for Klocwork 2020.1	2020-03-12
ABedford	5.1	Updated for Klocwork 2020.2	2020-06-29
ABedford	5.2	Updated for Klocwork 2020.3	2020-09-14
ABedford	5.3	Updated for Klocwork 2020.4	2021-02-24
ADunster	6.0	Updated for Klocwork 2021.1	2021-04-26
ADunster	6.1	Updated for Klocwork 2021.2	2021-08-17

ADunster	6.2	Updated for Klocwork 2021.3	2021-11-30
ADunster	6.3	Updated for Klocwork 2021.4	2022-01-18
JBritton	6.4	Updated for Klocwork 2022.2 and Klocwork 2022.1 plus rebranding	2022-06-30
JBritton	6.5	Updated for Klocwork 2022.3 and Klocwork 2022.4	2022-12-17
JBritton	6.6	Updated for Klocwork 2022.4.SR1	2023-04-25
JBritton	6.7	Updated for Klocwork 2023.2	2023-07-26
JBritton	6.8	Updated for Klocwork 2023.4	2023-12-29

Contents

Release Notes Klocwork 2023.4	6
What's new in Klocwork 2023.4.....	6
Fixed issues in Klocwork 2023.4.....	20
Limitations	26
Release Notes Klocwork 2023.3	42
What's new in Klocwork 2023.3.....	42
Fixed issues in Klocwork 2023.3.....	50

Referenced Standards

Standards referenced in this document refer to the following versions:

Standard	Version
ISO 26262	ISO 26262:2018
IEC 61508	IEC 61508:2010
IEC 62304	IEC 62304:2006/AMD1:2015
EN 50128	EN 50128:2011/A2:2020

Trademarks

"MISRA", "MISRA C" and "MISRA C++" are registered trademarks of The MISRA Consortium Limited.
Windows is a registered trademark of Microsoft Corporation.

Related Documents

Document ID	Title
KW2023_4_001	Functional Safety Manual for Klocwork
KW2023_4_003	Klocwork ISO 26262 / IEC61508 / EN 50128 Certified Checkers
KW2023_4_005	Klocwork Checker Qualification Pack

Release Notes Klocwork 2023.4

These release notes cover Klocwork 2023.4 and include information about what's new in this release, issues we've fixed since the last release, and any limitations you should be aware of.

(Also available at <https://help.klocwork.com/current/en-us/concepts/releasenotes.htm>)

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

2022 licenses are not compatible with Klocwork 2023.4. You need a new license to use the latest version of the product. Contact license@perforce.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 2023.4

Here are the highlights for Klocwork 2023.4. If you're upgrading, see the [Limitations](#) for items that affect how you use Klocwork. (also available at <https://help.klocwork.com/current/en-us/concepts/whatsnewmain.htm>)

Command options for exact match and overrides file

In this release, we have introduced new functionalities for exact match and overrides file options in both `kwcheck` and `kwciagent`.

- **Exact File Matching:**

Enable exact file matching by setting the `enable_exact_file_match` option to true when using `kwcheck` or `kwciagent`.

- **File Overrides:**

Run the `--overrides-file` option with `kwcheck` or `kwciagent` to apply file matching overrides.

Clean command for `kwcheck` and `kwciagent`

In this release, we have introduced the command "clean" in both `kwcheck` and `kwciagent` to clean your local Klocwork project and/or settings directories.

Deprecation of `docs.roguewave.com` in 2024

The `docs.roguewave.com` site will be deprecated in early 2024. Refer to the offline help documentation that is included with the product for versions 2021 and earlier.

Klocwork/Validate Server

In this release, Apache Tomcat has been upgraded to version 8.5.96, with enhanced performance, security features, and additional optimizations for a more efficient and reliable server environment.

Plugins and extensions

From Klocwork 2023.4, the Klocwork Desktop Analysis plugins for IntelliJ IDEA, Android Studio, and CLion will work when using version 2023.1 or higher of the IDEs.

In release 2023.4, stability fixes have been completed for the Klocwork extension for Visual Studio Code and Klocwork Desktop plug-in for Visual Studio. Several customer issues have also been fixed, including:

- In Visual Studio Code, the error notification on startup will appear only if the workspace contains Klocwork analysis folders.
- In Visual Studio Code, the status options are ordered consistently in different areas of the extension.
- In Visual Studio, the infobar will be enabled immediately and updated after Klocwork options are saved.
- In Visual Studio, the defects of server taxonomies will disappear after being disconnected from the server.

C/C++

In this release we

- improved tracking of array values when using constant indices

- improved the desktop analysis tools (kwcheck/kwciagent) to match more closely the results of the server analysis tool (kwbuildproject).
- improved the C/C++ analysis engine for stability and accuracy
- significantly updated the HKMC taxonomy for more accurate coverage of rules

C#

In this release we

- added a new taxonomy for CWE 2023 Top 25 for C#

Java

In this release we

- added a new taxonomy for CWE 2023 Top 25 for Java
- added support for Gradle 8.3
- improved CERT Java coverage
- expanded analysis and identification of potential issues for enhanced code security

Coding standards

This release includes new and expanded standards coverage for the following coding standards:

- CWE IDs and 2023 Top 25 for C/C++, C#, and Java
- CERT Secure Coding Standard
- DISA STIG
- HKMC Secure Coding Standard
- ISO/IEC TS
- Klocwork Quality
- MISRA 2012 AMD3
- Payment Card Industry Data Security Standard

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
CXX.SUSPICIOUS_INDEX_CHECK	This C/C++ checker detects when a suspicious index check is present before accessing an array at a specific index.
CXX.SUSPICIOUS_INDEX_CHECK.CALL	This C/C++ checker detects when a suspicious index check is present before accessing an array in another function.
CXX.SUSPICIOUS_INDEX_CHECK.ZERO	This C/C++ checker detects when a suspicious index check against zero is present before accessing an array, but the index value is not checked against the upper array boundary.
MISRA.BITFIELD.UNION	This MISRA checker detects when a union contains bit field(s) as member(s).
MISRA.INTEGER_CONSTANT.MACRO.FLOAT_VALUE	This MISRA checker detects when the argument of an integer-constant macro is not a floating-point number.
MISRA.INTEGER_CONSTANT.MACRO.RANGE	This MISRA checker detects when the argument of an integer-constant macro is not within the range.
MISRA.INTEGER_CONSTANT.MACRO.SUFFIX	This MISRA checker detects when the argument of an integer-constant macro is not an unaffixed integer constant.

Modified checkers

Checker	Description
ABV checkers	Overall improvements to the checkers
ABV.GENERAL	Updated the documentation and reduced false positives

Checker	Description
AUTOSAR.OP.BINARY.RETVAL	Reduced false positives
CS.HIDDEN.MEMBER checkers	Reduced false positives
CS.RLK	Reduced false positives
CWARN.IMPLICITINT	Reduced false positives
CXX.ERRNO.NOT_CHECKED	Updated the documentation with sample codes
CXX.SUSPICIOUS_INDEX_CHECK.CALL	Reduced false positives
INFINITE_LOOP.GLOBAL	Reduced false positives
INVARIANT_CONDITION.GEN	Updated the documentation with clarification
MISRA.ASSIGN.OVERLAP	Reduced false positives
MISRA.FLOAT_EQUAL	New defects detected
MISRA.MEMB.NON_CONST	Reduced false positives
MISRA.ONEDEFRULE.VAR	Reduced false positives
MISRA.VAR.HIDDEN	Reduced false positives
MLK checkers	Reduced false positives
NNTS.MUST	Overall improvements to the checker
NNTS.TAINTED	New defects detected
NPD checkers	New defects detected
RABV.CHECK	Reduced false positives

Checker	Description
RLK checkers	Reduced false positives
RN.INDEX	Reduced false positives
SV.STRBO.BOUND_COPY.OVERFLOW	New defects detected
SV.STRBO.NOBOUND_COPY	New defects detected
SV.TAINTED.CALL.INDEX_ACCESS	New defects detected
SV.TAINTED.GLOBAL	New defects detected
UNINIT.CTOR.MUST	Reduced false positives
UNUSED.FUNC.GEN	Reduced false positives
UFM checkers	New defects detected

Enabled or disabled checkers

No checkers were added to or removed from 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	New/updated
cert_cpp.tconf and cert_c_pp_ja.tconf	Added or modified checker mappings to the following rules:

Taxonomy	New/updated
	<ul style="list-style-type: none"> ▪ CERT EXP56-CPP
cert_java.tconf and cert_java_ja.tconf	<p>The cert_java.tconf and cert_java_ja.tconf taxonomies were renamed from cert_java_community.tconf and cert_java_community_ja.tconf, respectively.</p> <ul style="list-style-type: none"> ▪ Added a substantial number of mappings to the cert_java.tconf and cert_java_ja.tconf taxonomies.
cwe_2019_top_25_cxx.tconf and cwe_2019_top_25_cxx_ja.tconf cwe_2020_top_25_cxx.tconf and cwe_2020_top_25_cxx_ja.tconf cwe_2021_top_25_cxx.tconf and cwe_2021_top_25_cxx_ja.tconf cwe_2022_top_25_cxx.tconf and cwe_2022_top_25_cxx_ja.tconf cwe_2023_top_25_cxx.tconf and cwe_2023_top_25_cxx_ja.tconf	<p>Added or modified checker mappings to the following weaknesses:</p> <ul style="list-style-type: none"> ▪ CWE-119 ▪ CWE-125 ▪ CWE-787

Taxonomy	New/updated
cwe_2023_top_25_cs.tconf and cwe_2023_top_25_cs_ja.tconf cwe_2023_top_25_java.tconf and cwe_2023_top_25_java_ja.tconf	<ul style="list-style-type: none"> ▪ Added new taxonomies that map Klocwork checkers to the 2023 CWE Top 25 Most Dangerous Software Weaknesses.
cwe_all_cxx.tconf and cwe_all_cxx_ja.tconf	Added or modified checker mappings to the following weaknesses: <ul style="list-style-type: none"> ▪ CWE-119 ▪ CWE-124 ▪ CWE-125 CWE-787
disa_stig_v4_cxx.tconf and disa_stig_v4_cxx_ja.tconf disa_stig_v5_cxx.tconf and disa_stig_v5_cxx_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> ▪ APSC-DV-002590 ▪ APSC-DV-003170
disa_stig_v5_java.tconf and disa_stig_v5_java_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> ▪ APSC-DV-001540

Taxonomy	New/updated
hkmc_c.tconf and hkmc_c_ja.tconf hkmc_cpp.tconf and hkmc_cpp_ja.tconf	<ul style="list-style-type: none"> ▪ Substantial reorganization of the hkmc_c.tconf and hkmc_c_ja.tconf taxonomies.
iso_iec_ts_17961_c.tconf and iso_iec_ts_17961_c_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> ▪ 5.22
kw_quality_std_cxx.tconf and kw_quality_std_cxx_ja.tconf	Added or modified checker mappings to the following categories: <ul style="list-style-type: none"> ▪ Buffer Overflow
misra_c_2023_c11_all_checkers.tconf and misra_c_2023_c11_all_checkers_ja.tconf misra_c_2023_c11_certified.tconf and misra_c_2023_c11_certified_ja.tconf	Added or modified checker mappings to the following categories: <ul style="list-style-type: none"> ▪ Mandatory Rules Required Rules
misra_c_2023_c90_all_checkers.tconf and misra_c_2023_c90_all_checkers_ja.tconf misra_c_2023_c90_certified.tconf and misra_c_2023_c90_certified_ja.tconf	Added or modified checker mappings to the following categories: <ul style="list-style-type: none"> ▪ Required Rules

Taxonomy	New/updated
<p>misra_c_2023_c99_all_checkers.tconf and misra_c_2023_c99_all_checkers_ja.tconf</p> <p>misra_c_2023_c99_certified.tconf and misra_c_2023_c99_certified_ja.tconf</p>	<p>Added or modified checker mappings to the following categories:</p> <ul style="list-style-type: none"> ▪ Mandatory Rules ▪ Required Rules
<p>pci_3_2_1_cxx.tconf and pci_3_2_1_cxx_ja.tconf</p>	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> ▪ 6.5.2

Improvements to supported compilers

We've added or improved support for the following compilers:

- Clang
- Clang-cl
- GNU
- Green Hills
- Microsoft Visual C++
- Windriver GCC

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

Licensing

Klocwork supports Reprise License Manager (RLM).

2022 licenses are not compatible with Klocwork 2023.4. You need a new license to use the latest version of the product. Contact license@perforce.com to obtain a new license.

In this release, we have added information about using RLM dongles with Klocwork. For more information, see [Supported versions of RLM](#) and [Operating systems that support RLM dongles](#).

End of Life notice for FLEXlm/FlexNet Publisher as of Klocwork 2023.1

Klocwork has changed its license management tool by moving from FLEXlm/FlexNet Publisher to Reprise License Manager (RLM) as of Klocwork 2023.1. FLEXlm/FlexNet Publisher is no longer supported.

New product license files will be generated for Reprise; if you require a FLEXlm license file for older Klocwork versions, we can provide this for you.

To learn more about transitioning, see [Transition license from FlexLM to Reprise](#).

Changes to system requirements

In this release, we've added support for

- Windows 11 (version 23H2)
- Debian 11.8
- Amazon Linux 2 (2.0.20231101.0 Update)

- Ubuntu 20.04.6 LTS, 22.04.3 LTS
- openSUSE Leap 15.5
- SUSE Enterprise 15 SP5
- Eclipse 4.29 (2023-09)
- Android Studio Giraffe (2022.3.1 Patch 3)
- Visual Studio 2017 version 15.9.58
- Visual Studio 2019 version 16.11.31
- Visual Studio 2022 version 17.7.6
- Visual Studio Code 1.84.1 (minimum supported version is 1.74.3)
- IntelliJ IDEA 2023.1.5, 2023.2.4
- CLion 2023.2 (up to 2023.2.2)
- Microsoft Edge 108.x to 119.x
- Firefox 119.x, 115.x ESR
- Chrome 119.x
- Jenkins 2.431
- Gradle 8.3
- Windows RLM v15.1BL2
- Linux RLM v15.1BL2

In this release, we've ended support for

- Ubuntu 16.04
- Visual Studio 2022 versions 1.72.2 to 1.74.2
- IntelliJ IDEA 2019.1, 2019.2
- Microsoft Edge 105.x to 107.x
- Firefox 105.x to 106.x
- Chrome 106.x to 108.x

For the complete list of supported versions, see [System Requirements](#).

Discontinuation of Klocwork Server installations in release 2023.4

Starting from release 2023.4, Klocwork Server installations have been discontinued. We recommend transitioning to [Validate installation](#) for a more streamlined and integrated experience.

When transitioning from Klocwork to Validate:

- Stop your Klocwork instance and back up the projects_root.
- During Validate install, set the projects_root location to your current projects_root.
- If you are currently using non-default values for ports or license server, be sure to set the same values during Validate install.
-

Deprecation of issue grouping

Issue grouping is deprecated as of Klocwork 2023.3. If you are upgrading from a previous version, we recommend turning off issue grouping before performing a migration.

Maintenance for Klocwork 2021 ended

Maintenance for all versions of Klocwork 2021 ended March 31, 2023. The end of maintenance (EOM) date and end of sale (EOS) date was also March 31, 2023. For information about the availability of support for any release of Klocwork, see the Klocwork Product Lifecycle.

Path API version upgrade in Klocwork 2023.1

We upgraded the Path API version to accommodate multi-threaded execution within path analysis instances. The upgraded API is not backward compatible with previous versions. All custom checkers using the Path API need to be

updated and recompiled by using the 2023 Klocwork Path API headers and library. To learn more, see the Path API documentation.

[End of Life notice for macOS as of Klocwork 2023.1](#)

Beginning with Klocwork 2023.1, the following operating systems and installers are not supported:

- macOS

Fixed issues in Klocwork 2023.4

The following issues were fixed in Klocwork 2023.4.

(also available at <https://help.klocwork.com/current/en-us/concepts/fixedissues.htm>)

General issues

Number	Description
SUPPORT-41257	Fixed an issue with INFINITE_LOOP.GLOBAL not being detected in Modern Engine.
00637860, SUPPORT-44290	Updated the documentation for the checker CXX.ERRNO.INCORRECTLY_CHECKED by adding code examples.
00603314, SUPPORT-48576	Updated the kwsync --dry option output to clearly indicate the change that would be applied and the source and destination projects.
00661525, 00850370, 00978708, SUPPORT-49136	Improved support for the Klocwork Web API to allow users to save compliance reports externally.
00982409	Fixed an issue related to java.lang.IndexOutOfBoundsException on Android 12 and 13.

Number	Description
00816426, 00894852, 001021775	Improved support for the GNU compiler to address an issue with C/C++ analyzer stability.
00831709, 00861988	Updated the options in the Taxonomy dropdown menu for Compliance Reports in Validate.
00846054	Updated documentation for the 2022.3 release notes with checkers that were enabled by default.
00854423, 00875348, 00992578, 01000428	Fixed an issue with auto-delete of previous builds failing during parallel load to streams.
00891330, 00999245, 00988104, 01012137	Fixed an issue with compiler version errors appearing during a build integration.
00886440	Fixed an issue with uploading Java analysis tables to the Klocwork server on Linux.
00925447	Fixed an issue related to CI builds not uploading when the mconf file is considered for integration builds.
00944853	Updated the documentation for filtering out issues using .sconf files.
00931333	Improved support for the Clang compiler.

Number	Description
00984993	Updated the checker mappings for the HKMC C and C++ taxonomies.
00989686, 001006425	Fixed an issue with Validate not loading results on the Projects page due to invalid project names.
00996396	Updated the documentation for the command line arguments related to the Klocwork desktop installer.
01004742	Added support for Gradle 8.3 for Java analysis.
01006281	Fixed issues related to generating a build specification for projects based on Android 13.
01008727	Fixed an issue with an error appearing when installing Validate.
01014082	Updated the documentation for the checker INVARIANT_CONDITION.GEN.
01017758	Updated the documentation with options for kwciagent run.
01020100	Updated the documentation with enable_exact_file_match property.
01020562	Updated the documentation with the number of bytes allowed for build tags.

Number	Description
01023487	Updated Apache Tomcat version to 8.5.96.
01027199	Fixed an issue with kwinject where compiler configuration required the option --debug to succeed.

Checker issues

Number	Description
01029331	Added the new checkers CXX.SUSPICIOUS_INDEX_CHECK, CXX.SUSPICIOUS_INDEX_CHECK.CALL, and CXX.SUSPICIOUS_INDEX_CHECK.ZERO.
SUPPORT-8703, SUPPORT-8719	Reduced false positives for the checker MISRA.ASSIGN.OVERLAP.
00687597	Reduced false positives for the checker ABV.GENERAL.
00710330, 00710333	Improved defect detection for the checkers SV.STRBO.NOBOUND_COPY and SV.STRBO.BOUND_COPY.OVERFLOW.
00777120, 00710367	Improved defect detection for the checker MISRA.FLOAT_EQUAL.

Number	Description
00849605	Reduced false positives for the checker INFINITE_LOOP.GLOBAL.
00820046	Reduced differences in results between local and integration build analysis.
00947481	Reduced false positives for the checker MISRA.ONEDEFRULE.VAR.
00943250	Reduced false positives for the checker UNINIT.CTOR.MUST.
00947481, 01003526	Reduced false positives for the checker UNUSED.FUNC.GEN.
00990095	Reduced false positives for the checker AUTOSAR.OP.BINARY.RETVAL.
00990095	Reduced false positives for the checker CWARN.IMPLICITINT.
00997581	Improved defect detection for the checker ABV.GENERAL.
00995937	Improved defect detection for the checker NNTS.TAINTED.
01006582, 01006571	Improved defect detection for the UFM checkers.

Number	Description
001027143	Reduced false positives for the checkers MISRA.MEMB.NON_CONST and MISRA.VAR.HIDDEN.

Limitations

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

(also available at <https://help.klocwork.com/current/en-us/concepts/kwlimitations.htm>)

Limitations for installation, upgrade, and deployment

Previous dbvalidate and dbvalidate cleanup versions should not be used with previous Validate versions that have CI builds

Using the 2023.1-2023.3 versions of dbvalidate and dbvalidate cleanup with projects-root that have CI builds can result in data loss.

Workaround: To run dbvalidate and dbvalidate cleanup before upgrading from Validate versions 2023.1-2023.3, use the versions included in the 2023.4 Validate installation package.

Klocwork desktop tools cannot connect to previous servers

The 2023.4 Klocwork desktop tools cannot connect to previous (23.3 and earlier) Klocwork servers, due to internal changes in the tools and the server APIs they communicate with.

On Windows, integration build results fail to load to the Server

On Windows, when installing Klocwork, do not choose the **Custom** setup type and select the **Klocwork Servers Only** option, because integration build results will fail to load to the Server.

Workaround: Install Klocwork and choose either the **Complete** setup type that installs all program features. Or, choose the **Custom** setup type and select the **Klocwork Servers and Built Tools** option.

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.

kwcollect fails on tables generated by new 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.

Debian 10.x and later cannot run the license server

Debian no longer supports the Linux Standard Base core (lsb-core) as of version 10.x. Therefore, the license server cannot be run on the Debian 10.x and later platforms.

kwauth doesn't properly set HTTP/1.1 header

Sometimes when the Klocwork Server IP address is associated with multiple host names or located behind a reverse proxy, kwauth does not properly resolve the FDQN of the Klocwork Server.

Workaround: To resolve this problem, we added a conditional host resolution based on a parameter in a specified configuration file. If you set it to 'false', then you can specify FQDN for the URL of the remote server. To set host resolution to 'false', you need to create a configuration file on the client side with the following address:

```
{client_tools_install_folder}\config\client_config.xml
```

The file must have the following structure:

```
<?xml version="1.0" encoding="UTF-8"?><params>    <host resolveHost="false" /></params>
```

64-bit REPCXX Memory Limitations

With 2020.4 SR1, 64-bit REPCXX (our classic C/C++ analysis engine) by default is capped at using 16 Gigabytes (GB) of memory. An attempt to exceed this limitation will result in the REPCXX process being terminated with exit code 121 and the following message:

```
"memory: REPCXX has exceeded the memory threshold of 16000000 Kb, mem_usage={0} Kb.  
Memory threshold can be increased by passing a value (in GB) to REPCXX through the  
'--memory-limit=' option or through the 'KW_CLASSIC_ENGINE_MEMORY_LIMIT' environment  
variable."
```

Workaround: As the error message indicates, the new --memory-limit REPCXX option or 'KW_CLASSIC_ENGINE_MEMORY_LIMIT' environment variable may be used to override the 16 GB memory limit. For example, to increase the memory limit to 32 GB per REPCXX process, you can run kwbuildproject with option --add-compiler-options '--memory-limit=32'.

As confirmation of the memory limit increase, the following message will be output to the build log when a REPCXX process is initiated:

```
"REPCXX memory input set to 32 GB (32000000 KB)"
```

'kwloaddb: Java heap space' error when trying to load a build

During a load or a build from a connected project, or when running kwcheck run, an 'OutOfMemory' exception may occur.

Workaround: To fix this error, you can try to increase the amount of memory for kwloaddb in your java_wrappers_memory.conf file. If you continue to see the error, contact [Customer Support](#).

dbvalidate outputs error(s) when attempting to validate a database that contains streams

For projects with streams, when attempting to run dbvalidate as follows:
`java -jar /<path-to-server>/class/dbvalidate.jar --projects-root /<path-to-projects-root>/projects_root --project a`

dbvalidate will incorrectly output error messages on steps a1, k, and occasionally j and l. These messages can be ignored.

Using 'sconf' files on Linux does not filter defects in the file specified.

If you are using windows style paths in an 'sconf' file, connected projects can yield inconsistent results.

Workaround: Avoid using of windows style file paths (For example, 'C:\Checkout\PRs\PR-59822\test.cpp' or 'C:\\Checkout\\PRs\\PR-59822\\test.cpp'). You can use UNIX-style paths across all platforms, for example:

Windows: `C:/Checkout/PRs/PR-59822/test.cpp`

Linux: `/checkout/mydir/test.cpp`

Note: Use of the 'C:\Checkout\PRs\PR-59822\test.cpp' format should be avoided on **all platforms** as it is not supported.

Limitations for build integration

Cannot load Android 4.4 (KitKat) using the default memory settings for kwloadadb, 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

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

Feature	Details
Build integration	<ul style="list-style-type: none">▪ Build specification templates
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:

- Display of server issues in Visual Studio
- File-level analysis in [Visual Studio](#) (only solutions and projects can be analyzed)
- Using [knowledge bases](#)
-

Limitation for Python

In release 2023.3, the following checkers listed in the Python taxonomy are not supported. They are disabled by default and can be enabled manually:

0001	F0001	I0001	I0020
E0013	F0002	I0010	I0021
E0014	F0010	I0011	I0022
E0015	F0011	I0013	I0023

In release 2023.2, the following checkers listed in the Python taxonomy are not supported. They are disabled by default and can be enabled manually:

E0001	F0002	I0020	W4902
E0013	F0010	I0021	W4903
E0014	F0011	I0022	W4904
E0015	I0001	I0023	W4905
E0705	I0010	W0246	
E1143	I0011	W3101	
F0001	I0013	W4901	

Limitations for Klocwork Static Code Analysis/Validate

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.

The 'Cert C Recommendations' section of the Security Report is omitted.

Workaround: You can generate the compliance report for this taxonomy if needed. Contact support for more information.

Cannot run Klocwork Server and Validate Windows services at the same time

On Windows, if you install either of the Validate or Klocwork Server installations, and set the Windows services to start automatically, then choose to additionally install the other installation, the services will not start. You can run the Windows services for only one of these installations at a time.

Additionally, make sure if you're installing both installations, their web ports and database ports must be different.

Workaround: If you want to run both simultaneously, you must start the servers for at least one installation by using the command line.

CI build paths are not shown in the tree structure of the 'Modules' page in Klocwork Static Code Analysis/Validate

When creating a new module, the file tree always shows files from the latest Server build (even if the latest uploaded build was CI build from kwciagent).

Workaround: If you want to include files or paths from a CI build, you need to manually input them using 'Path patterns' instead of using the 'Use tree' field.

An error occurs when attempting to 'Edit All' issues in a project with CI Builds only

When citing issues in Validate from the project's issue list, the following error occurs if you attempt to use **Edit All** to manage your issue list if your project only contains CI Builds:

Database error occurred: There is no valid build for the project

Workaround: Select the issues you want to cite and use **Edit Selected**.

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 extension

'One or more extensions were loaded using deprecated APIs' warning message in Visual Studio 2019

Visual Studio 2019 may give a warning message regarding deprecated APIs. If you select the recommended option to not allow deprecated API usage, this will disable the Klocwork plug-in and you will no longer be able to access the Klocwork tools in VS.

Workaround: Select the 'Don't show this message for current extensions' option to safely ignore this warning and continue to use the Klocwork plug-in.

'Visual Studio stopped responding for X seconds.' warning message in Visual Studio 2019

Visual Studio 2019 may give a warning message regarding slower performance in relation to use of the Klocwork plug-in. If you select the option to 'disable this extension', it will disable the Klocwork plug-in and you will no longer be able to access the Klocwork tools in VS.

Workaround: Select the 'Don't show this message for current extensions' option to safely ignore this warning and continue to use the Klocwork plug-in.

Visual Studio 2015 preventing analysis of C++

A known issue with Visual Studio 2015 running on Windows Server 2012 prevents the Klocwork extension for Visual Studio from analyzing C++ projects.

Workaround: Either modify the build configuration version to x86, or switch the analysis mode to Classic. To switch to Classic mode, go to **Tools > Extensions and Updates > Klocwork > Options**, and select the **Analysis** tab. Under **Analysis Mode**, select the Use Classic Mode check box.

Visual Studio extension may be disabled

When installing the Klocwork extension for Visual Studio into Visual Studio 2015, the extension may initially be disabled. If so, go to **Tools > Extensions & Updates** and select **Installed**. Select the **Klocwork Extension for Visual Studio**. You may need to restart Visual Studio.

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.

'kwcc' error in Visual Studio after upgrading

If, before 2020.1, you deployed the MISRA checkers to your project using kwdeploy, and have a misra.xml file in your %USERPROFILE%\klocwork\plugins folder, you might see errors similar to the following:

```
kwcc: Error: C:\Users\username\.klocwork\plugins\misra.xml:5783:
Trying to describe error 'MISRA.STDLIB.ILLEGAL_WRITE.2012_AMD1' several times.
Repeated descriptions are ignored
```

Workaround: To fix this issue, delete the misra.xml file located in your %USERPROFILE%\klocwork\plugins folder before performing the upgrade.

Help for Klocwork community checkers cannot be accessed directly from Visual Studio

If you attempt to access the help for a community checker by right-clicking the checker and selecting **View Checker Documentation**, you will get a 'Cannot find requested topic on your computer' error message.

Workaround: Offline help for the community checkers is available by using the portal. [Online help](#) is also available.

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).

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 try to open help for an issue from the Klocwork Issues window in Visual Studio for the Klocwork extension for Visual Studio

If you click an issue in the Klocwork Issues window and try to open the help 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.

Options dialog is unresponsive if the license host box contains an invalid character

If you enter the details for your license host and type an invalid character (such as a ';' or '#'), the dialog box turns red and you can't change any other settings within the Options dialog box.

Workaround: To fix this issue, replace the content of the license host box with a valid character (any letter of the alphabet).

Defect column does not update correctly unless tabs are converted to spaces

If you press **tab** to indent code, this will insert a tab character or a configurable number of spaces. When the Klocwork extension shows the column number where a defect occurs, it counts a tab as one space. If you do not configure Visual Studio to convert tabs to spaces on source files that have tabs and when you press the **tab** key, the correct column information for defects may be incorrect.

Workaround: To ensure that Klocwork counts the correct number of columns when it displays where a defect has occurred, go to **Tools > Options > Text Editor > All Languages > Tabs** and select **Insert spaces**.

Klocwork Visual Studio plugin installation does not detect Visual Studio 2015

If you try to install the Klocwork Visual Studio plugin on Visual Studio 2015 and you have a higher version of Visual Studio (such as 2017 or 2019) already installed, the VSIX installer does not automatically detect the Visual Studio 2015 installation. The error "This extension is not installable on any currently installed products" appears.

Workaround: To install the Klocwork plugin on Visual Studio 2015, run the compatible VSIX installer from the command prompt:

<path-to-Visual-Studio-2015-compatible-VSIX-Installer> \VSIXInstaller.exe kw-vsplugin.win64.vsix

```
"C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\VSIXInstaller.exe"  
C:\KW_Installer\23.1\kw-desktop-tools.23.1.0.61.win64\kw-vsplugin.win64.vsix
```

Limitations for VS Code

Do not add the build specification location to the build command. It will be automatically collected from the 'Build Specification Location' setting.

'Build Specification Generation Command' should not contain the buildspec location (-o <location>) when using the 'Automatically Update Build Specification' option.

Workaround: Specify the location with the Build Specification Location setting.

Build Specification Generation Command cannot contain single quoted arguments.

If you need quoted parameters, use double quotes (same as a command prompt/terminal).

Workaround: Ensure your quoted parameters use double quotations.

Citing defects only works when using a connected project.

If your project is not connected and you attempt to cite defects, nothing will happen.

Workaround: Ensure that all 'Connection Settings' (host, port, SSL, license host, port, and Klocwork project) are set, and the Portal instance the settings are pointing to is available and running.

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.

Limitation for the Eclipse plug-in

Eclipse 4.16 requires additional software for Klocwork Eclipse plug-in to run

If you are using the 4.16 version of Eclipse, you must install the following software in order to run the Klocwork Eclipse plug-in:

- **Eclipse CDT** using Java 11; or, post-installation, update the eclipse.ini file to point to a valid Java 11 installation.
- **Eclipse Java Development Tools** by using the Help --> Install New Software command.
-

Limitation for the IntelliJ IDEA plug-in

Difficult to connect to a project or stream with a long name

If you want to connect to a project or stream that has a long path or name, you may not see the full path of the project or stream because the Klocwork Project dropdown list might be too narrow to display the entire path and name.

Workaround: You can fix this problem by doing the following:

1. Open your IDE with Klocwork installed and then open a project. Doing so ensures the .idea folder gets generated in your project.
2. Close your IDE.
3. In your project's .idea folder, edit the 'misc.xml' file. Edit the following line by adding the project/stream name in the value parameter (The name appears with slashes included if it's a stream):
`<option name="kwProject" value="project-name/stream-name" />`
4. Open your IDE. If you open your Klocwork settings, you can verify that you are connected to the correct project or stream.

WARN: Do not use URL connection as JarURLConnection' message in IntelliJ IDEA log

If you see this warning in IntelliJ IDEA in your idea.log, you can safely ignore it.

Limitation for the CLion plug-in

'java.sql.SQLException: SQLite.Exception: no such column: in_connected' error

This error occurs if you are trying to use a 2022.1 or newer version of the Klocwork Desktop plugin for CLion with a 2021.4 or older version of the Klocwork tools.

Workaround: If you are still using an older version (2021.4 or older) of the Klocwork tools, do not upgrade the Klocwork CLion plugin to 2022.1 or newer to avoid this error.

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. On Windows, the makefile generated by kwcreatechecker requires nmake to build the checker.

Workaround: None.

Checker limitations on Windows

kwcreatechecker.exe creates 64-bit checkers by default. If you want to create 32-bit checkers (that is, checkers compatible with a pre-2020.4 version of Klocwork) you must use the --force-32bit option.

If you use pre-2020.4 KAST / AST checkers or pre-2020.3 PATH checkers, you must add the --force-32bit option to all builds (kwbuildproject, kwcheck, and so on).

If you use pre-2020.4 KAST / AST checkers or pre-2020.3 PATH checkers and want to create new checkers, you must do one of the following:

- Rebuild all your old checkers by using the 2020.4 version of kwcreatechecker.exe or by using a 64-bit compiler.
- Build your new checkers in a pre-2020.4 version.

You cannot mix 32-bit and 64-bit checkers:

- If you build any 32-bit checkers, you can only use them in the 2020.4 Visual Studio plugin if the Force 32-bit Analysis option is enabled.
- If you build any 64-bit checkers, you can only use them in the 2020.4 Visual Studio plugin if the "Force 32-bit" Analysis option is disabled.
- If you build any 32-bit checkers, you cannot use them in Eclipse-based plugins.
- When using 64-bit path checkers, you will get the following warning (note that it is safe to ignore this warning):

```
kwcc: Warning: cannot open plugin library 'c:\Klocwork Server\plugins\ix86-pc-win32\&lt;checker>.dll': The specified module could not be found.
```

Checker limitations on Linux

32-bit backward compatibility for custom checkers is no longer supported and the option '--force-32bit' is deprecated. You must rebuild all of your old checkers by using a 64-bit compiler. Contact support for more information.

Custom checker help is not found when searching in offline help

As of 2022.1, it is not possible to search for custom checker help from the offline documentation accessed from the Klocwork Portal.

Workaround: You can search for custom checker help from the Configuration tab in the Klocwork Portal instead. From your project, click **Configuration**, select any taxonomy, and use the search box above the checker list. The help for any custom checker appears in the right pane.

Custom checker help cannot be displayed in some IDEs

Custom checker help does not display for Eclipse, Klocwork Desktop, IntelliJ IDEA, and Android Studio.

Workaround: You can view custom checker help by accessing it through the Configuration tab in the portal. From your project, click **Configuration**, select any taxonomy, and use the search box above the checker list. The help for any custom checker appears in the right pane.

Release Notes Klocwork 2023.3

These release notes cover Klocwork 2023.3 and include information about what's new in this release, and issues we've fixed since the last release,

What's new in Klocwork 2023.3

Here are the highlights for Klocwork 2023.3. If you're upgrading, see the [Limitations](#) for items that affect how you use Klocwork.

Identify builds with build tags

We have introduced build tags for both system (regular) builds and CI builds. Build tags brief elements of information that allow you to

- better identify specific builds, by using useful tags such as branch names, commit IDs, platforms, or architecture
- add arbitrary metadata to a build, that can be used in your scripts
- associate a commit or branch with a build, which can then be queried and used to help generate the file matching overrides file in an automated fashion

There are many ways to add, edit, and view build tags. They are accessible through the web Portal, the web API, kwadmin or validate admin's load & list-builds commands, and kwciagent's run and sync commands. To learn more, see [Using build tags](#).

Manually specify file matches using an overrides file

We have added a file matching overrides file option to manually specify file matches.

The overrides file is a simple text file that allows you to mark files as added, deleted, or renamed. Using an overrides file gives you a better control over complex scenarios and reduces file mismatches.

To apply the overrides file, use the file-overrides option with kwadmin load or validate admin load. See [Use file matching overrides file](#) for more information and examples.

Recommendations for loading builds

Klocwork uses an auto-matching algorithm when loading builds to determine which files have changed or moved when compared to previous builds. This is the foundation for matching issues between builds, and has a big impact on the accuracy of results build over build. In order to get the best matching results when loading Klocwork builds, we encourage you to:

- always use replace path
- enable exact file matching

- use a file matching overrides file
- use build tags
- create logical stream structures

For more information, see [Recommended mechanisms for loading builds](#).

Plugins and extensions

From Klocwork 2023.3, analysis done in Visual Studio will default to the kwcheck external analysis engine. Having kwcheck.exe as your default engine allows you to bypass indexing to start your analysis sooner, and lets you use the same underlying analysis framework both on the desktop and on the server, resulting in improved consistency and efficiency in resolving your defects.

In order to use the external engine, ensure that you have installed kwcheck. To configure kwcheck, navigate to **Extensions > Klocwork > Options**.

Streams

We've further enhanced the speed and performance when working with, editing, and deleting streams in a project.

C/C++

In this release we

- added tracking of array values when using constant indices
- improved the C/C++ analysis engine for stability and accuracy
- enhanced support for C++14 and C++17 analysis
- added a new taxonomy for CWE 2023 Top 25 for C/C++
- added a new taxonomy for MISRA C:2023

Java

In this release we

- added full support for Java 14 analysis

Coding standards

This release includes new and expanded standards coverage for the following coding standards:

- CWE
- CWE 2023 Top 25 for C/C++
- DISA-STIG for C/C++
- HKMC Secure Coding Standard
- MISRA C:2023

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
PRECISION.LOSS.INIT	This C/C++ checker detects when an implicit cast to a smaller data type during initialization may cause a loss of precision (data).

Modified checkers

Checker	Description
INFINITE_LOOP.GLOBAL	New defects detected
INFINITE_LOOP.LOCAL	New defects detected
LOCRET.ARG	Reduced false positives
MISRA.EXPANSION.UNSAFE	New defects detected
MLK.MUST	New defects detected
NPD.FUNC.MIGHT	Reduced false positives
RH.LEAK	Reduced false positives
RLK.OUT	New defects detected
SV.SSRF.URI	New defects detected
SV.USAGERULES.PROCESS_VARIANTS	Updated the documentation with sample codes
UNINIT.STACK.MUST	Reduced false positives
UNREACH.GEN	Reduced false positives
VA_UNUSED.INIT	Reduced false positives

Enabled or disabled checkers

The following checkers were removed from the default `enabled` field of the checker configuration files for this release.

- PY3.E0001
- PY3.E0013
- PY3.E0014
- PY3.E0015
- PY3.F0001
- PY3.F0002
- PY3.F0010
- PY3.F0011

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	New/updated
cwe_2023_top_25_cxx.tconf and cwe_2023_top_25_cxx_ja.tconf	<ul style="list-style-type: none"> ▪ Added new taxonomies that map Klocwork checkers to the 2023 CWE Top 25 Most Dangerous Software Weaknesses.
cwe_all_cxx.tconf and cwe_all_cxx_ja.tconf	<ul style="list-style-type: none"> ▪ Substantial reorganization of the cwe_all_cxx.tconf and cwe_all_cxx_ja.tconf taxonomies.

Taxonomy	New/updated
disa_stig_v4_cxx.tconf and disa_stig_v4_cxx_ja.tconf	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> ▪ APSC-DV-002590 ▪ APSC-DV-003170
disa_stig_v5_cxx.tconf and disa_stig_v5_cxx_ja.tconf	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> ▪ APSC-DV-002590 ▪ APSC-DV-003170 ▪ APSC-DV-002010
hkmc_c.tconf and hkmc_c_ja.tconf	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> ▪ C-INT-001 ▪ C-INT-003
misra_c_2023_c11_all_checkers.tconf and misra_c_2023_c11_all_checkers_ja.tconf misra_c_2023_c11_certified.tconf and misra_c_2023_c11_certified_ja.tconf	<ul style="list-style-type: none"> ▪ Added new tax

Taxonomy	New/updated
misra_c_2023_c90_all_checkers.tconf and misra_c_2023_c90_all_checkers_ja.tconf misra_c_2023_c90_certified.tconf and misra_c_2023_c90_certified_ja.tconf misra_c_2023_c99_all_checkers.tconf and misra_c_2023_c99_all_checkers_ja.tconf misra_c_2023_c99_certified.tconf and misra_c_2023_c99_certified_ja.tconf	

Improvements to supported compilers

We've added or improved support for the following compilers:

- Clang
- GNU
- TI Arm Clang
- TI tms320c28x
- TI msp430 C/C++
- TI tms320c6x, TI tms320c55x, and TI C7000 Optimizing C/C++

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

Changes to system requirements

In this release, we've added support for

- Windows 11 (version 22H2)
- AlmaLinux versions 9.0 to 9.2
- Amazon Linux 2 (2.0.20230808.0 Update)
- Oracle Linux 8.8
- Rocky Linux versions 9.0 to 9.2
- Eclipse 4.28 (2023-06)
- Android Studio Giraffe 2022.3.1
- CLion 2023.1 (up to 2023.1.5)
- Visual Studio 2017 version 15.9.56
- Visual Studio 2019 version 16.11.29
- Visual Studio 2022 version 17.7.1
- Visual Studio Code 1.81.1 (minimum supported version is 1.72.2)
- Microsoft Edge 115.x
- Firefox 116.x
- Chrome 116.x
- Jenkins 2.419

In this release, we've ended support for

- Visual Studio Code 1.69.1 to 1.72.1
- Microsoft Edge 103.x to 105.x
- Firefox 103.x to 105.x
- Chrome 103.x to 105.x

For the complete list of supported versions, see [System Requirements](#).

Fixed issues in Klocwork 2023.3

The following issues were fixed in Klocwork 2023.3.

General issues

Number	Description
SUPPORT-35237	Improved support for the Texas Instruments compiler for C2000 microcontrollers.
SUPPORT-36326	Updated documentation for upgrading from a previous version.
SUPPORT-35731	Updated documentation for moving a projects_root directory.
SUPPORT-37211	Updated documentation for issue search operations in Validate.
SUPPORT-38793	Fixed an issue with a missing drive root in the build specification include paths.
00624414, SUPPORT-42265	Updated the documentation for running the Klocwork Desktop Analysis (kwgcheck) on Debian 10.7.
SUPPORT-42382	Improved logged error messages when the --strict option is not specified.

Number	Description
SUPPORT-44002	Improved support for unqualified enums in the KAST API.
SUPPORT-48640	Fixed a parsing issue with C++ templates.
00905928, 00842180, 00807698, 00759012	Fixed an issue with the kwinject command using incorrect environment variable settings when configuring the Armclang compiler.
00859379	Added support for SELinux.
00947057	Added support for Windows 11.
00903613, 00904479	Fixed an issue with the Eclipse Update Site redirect.
00891330	Improved support for the Intel C++ Compiler 2022 with the Visual Studio plugin.
00925029, 00934307	Implemented a script to clean deleted builds from the Server.
00866775	Fixed an issue with the kwxsync/validate xsync --copy option not updating when the --status-priority option is used.

Number	Description
00924938	Fixed a cross-site scripting vulnerability issue identified in the Validate login page.
00712357	Fixed an issue related to an unsupported target in a C/C++ project.
00844364, 00922924, 00926713, 00920424, 00943099, 00994657	Updated the documentation with guidance about how to configure session timeouts to release open or expired sessions.
00902616	Fixed a permissions issue where a user who did not have permission to change issues statuses could select a new status from the drop-down list.
00944488	Fixed an issue with the kwant command not finding the jvm.dll from the Oracle JDK.
00811770	Fixed an issue related to parsing code that uses modern C++.
00904858	Fixed an issue with the kwxsync command not supporting synchronization between streams. Updated the documentation with a workaround to sync between empty base projects.
00891553	Fixed an issue with the web API accepting negative integers for the number of builds to keep.
00780020	Improved performance when managing and deleting a project with multiple streams.

Number	Description
00861363	Fixed an issue with a path pattern not parsing the + symbol correctly when defining modules in Validate.
00945027, 00891594, 00898942	Fixed an issue with the Report FP function related to creating an FP archive for an issue in a stream.
00975985	Updated the digital signatures for previous versions of the Klocwork and Validate installers and plugins.
00930450	Fixed an issue with a broken custom view of a taxonomy when performing a CI build sync.
00929145	Fixed an issue with modules not being updated dynamically when changing streams.
00943265	Improved the organization of the CWE C/C++ taxonomy.
00944618	Fixed an issue related to C/C++ analyser stability.
00934024	Improved support for the QNX compiler.

Number	Description
00975597	Updated the taxonomies and documentation to remove a small number of unsupported Python checkers.

Checker issues

Number	Description
00037123	Improved defect detection for the UFM.DEREF checkers for constructs involving arrays.
SUPPORT-37913	Improved defect detection for the checker INVARIANT_CONDITION.UNREACH.
00609990, SUPPORT-43031	Reduced false positives for the checker DBZ.GENERAL.
00640079, SUPPORT-43318	Improved defect detection for the NPD checkers.
00638935,	Improved defect detection for the checker MLK.MUST.

Number	Description
00683612, 00683625, 00711825	Improved defect detection for the checker ABV.GENERAL.
00862551, 00816520	Improved defect detection for the checkers INFINITE_LOOP.GLOBAL and INFINITE_LOOP.LOCAL.
00784688, SUPPORT-37314	Reduced false positives for the checker LOCRET.ARG.
00810594	Added the checker PRECISION.LOSS.INIT that reports a defect when an implicit cast to a smaller data type during initialization may cause a loss of precision.
00929566	Improved defect detection for the checker RLK.OUT.
00798227	Updated the documentation for the checker SV.USAGERULES.PROCESS_VARIANTS, clarifying that the checker reports issues on both Windows and Linux platforms.
00800042	Reduced false positives for the checker NPD.FUNC.MIGHT.
00757778	Improved defect detection for the checker SV.SSRF.URI not being reported on an Eclipse project. Updated the documentation with revised sample code to enable the checker.

Number	Description
00821309	Improved QDP tests for the checker MISRA.EXPANSION.UNSAFE.
00769992	Reduced false positives for the checker UNREACH.GEN.
00769996	Reduced false positives for the checker UNINIT.STACK.MUST.



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 exclusive property of RogueWave Software, Inc. a Perforce company. 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 Perforce Software, Inc. If you find any problems in the documentation, please report them to us in writing.

Klocwork is a registered trademark of RogueWave Software, Inc., a Perforce company.

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