



# Klocwork 2023.2 Release Notes

**Document Number:** KW2023\_2 \_009

**Certified version:** Klocwork 2023.2

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
<b>ABedford</b>	5.2	Updated for Klocwork 2020.3	2020-09-14
<b>ABedford</b>	5.3	Updated for Klocwork 2020.4	2021-02-24
<b>ADunster</b>	6.0	Updated for Klocwork 2021.1	2021-04-26
<b>ADunster</b>	6.1	Updated for Klocwork 2021.2	2021-08-17

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

## Contents

Release Notes Klocwork 2023.2 .....	6
What's new in Klocwork 2023.2 .....	6
Fixed issues in Klocwork 2023.2 .....	25
Limitations .....	32
Release Notes Klocwork 2023.1 .....	46
What's new in Klocwork 2023.1 .....	46
Fixed issues in Klocwork 2023.1 .....	63

## 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_2_001	Functional Safety Manual for Klocwork
KW2023_2_003	Klocwork ISO 26262 / IEC61508 / EN 50128 Certified Checkers
KW2023_2_005	Klocwork Checker Qualification Pack



## Release Notes Klocwork 2023.2

These release notes cover Klocwork 2023.2 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 2023.2. You need a new license to use the latest version of the product. Contact [license@perforce.com](mailto: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 2023.2

Here are the highlights for Klocwork 2023.2. 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>)

## Validate

We've made it easier for you to investigate issues in Validate. You can now view, modify, and navigate the issue search list without leaving the Issue Details page. Use the new File Navigation pane to search individual files and explore issues. These new UI panes are configurable and configurations persist between session when using the same browser.

You can now import and export QAC metrics in Validate. For a list of QAC metrics, see QAC metrics reference. We've improved the Metrics report designer: You can edit threshold and total-metric-value report definitions in Validate. We've also enhanced the report selection page.

We've updated the names of a few of the commands you can use with Validate. To learn more, see Validate command reference.

## Visual Studio extension

We've improved the performance of build specification generation for the Visual Studio extension when you use the kwcheck command as your external analysis engine. This means you'll be able to start analysis much sooner.

## Streams

We've dramatically reduced the time it takes to display and be able to use the project list for streams.

## C/C++

In this release we

- added support for several MISRA rules and increased coverage for MISRA C 2012 (up to AMD2).
- increased coverage for DISA STIG high severity rules

- improved coverage for CERT, OWASP, and CWE, including adding a taxonomy for the 2022 CWE Top 25 Most Dangerous Software Weaknesses
- enhanced support for C++14 and C++17 analysis

\*maximum coverage for MISRA C 2012 requires an additional package from Customer Support.

## C#

We improved support for version 8.0 of the C# language specification by

- adding support for static constructors
- adding support for nested types and operator declarations in interfaces
- improving support for verbatim interpolated strings
- improving support for ranges and indices by supporting the .. and ^ operators

## Java

We added

- support for Jakarta EE
- checkers that improve coverage of DISA STIG and CWE
- improved support for Java 14 switch expressions

## JavaScript

We added a --project-dir option to the kwjsspec command so that you can specify the JavaScript project root directory. This option helps capture the .eslintignore file (if present at the project root directory) for analysis.

## Coding standards



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

- CERT
- CWE
- DISA STIG
- MISRA
- OWASP

### 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.SV.PRIVATE_KEY.EMPTY_PASSWD	This C/C++ checker detects when an empty password is used to store a private key in a public key infrastructure (PKI) based authentication.
CXX.SV.PRIVATE_KEY.UNENCRYPTED	This C/C++ checker detects when an unencrypted cipher is used to store a private key in a public key infrastructure (PKI) based authentication, which can lead to unauthorized access.

Checker	Description
CXX.SV.PWD_INPUT.REVIEW	This C/C++ checker detects when password authentication is used in applications. Designers can review the defects to ensure that their company authentication policy is enforced.
CXX.SV.PWD.PLAIN	This C/C++ checker detects when an application attempts to set a password or PIN by using a string written in plain text.
CXX.SV.PWD.PLAIN.LENGTH	This C/C++ checker detects when an application attempts to set a plain text password that is fewer than 15 characters.
CXX.SV.PWD.PLAIN.LENGTH.ZERO	This C/C++ checker detects when an application attempts to set a plain text password that is zero characters long.
CXX.SIZEOF.CSTRING	This Community C/C++ checker detects the use of sizeof on a char*.
JAVA.SV.EMAIL.HOST	This Java checker detects cases where mail server hostname verification functions haven't been configured properly to ensure that a server presents the correct certificate.
JAVA.SV.XML.INVALID	This Java checker detects potential XML injection vulnerabilities by checking that XML data is XDS-validated before being processed.

Checker	Description
MISRA.MEMCMP.NTS.2012_AMD1	This MISRA checker provides support for MISRA 2012 AMD1: Rule 21.14 (Required): The Standard Library function memcmp shall not be used to compare null terminated strings.
MISRA.MEMCMP.NTS.GLOBAL.2012_AMD1	This MISRA checker provides support for MISRA 2012 AMD1: Rule 21.14 (Required): The Standard Library function memcmp shall not be used to compare null terminated strings.
MISRA.STDLIB.ABORT.2012_AMD2	This MISRA checker provides support for MISRA C 2012 AMD2 Rule 21.8: The library termination functions of <stdlib.h> shall not be used
MISRA.STDLIB.EOF.BAD_CMP.2012_AMD1	This MISRA checker provides support for MISRA 2012 AMD1: Rule 22.7 (Required): The macro EOF shall only be compared with the unmodified return value from any Standard Library function capable of returning EOF.

### Modified checkers

Checker	Description
ABV.GENERAL	New defects detected
ABV.STACK	Overall improvements to the checker
ANDROID.RLK.SQLOBJ	Reduced false positives
CS.HIDDEN.MEMBER.LOCAL.CLASS	Reduced false positives

Checker	Description
CS.HIDDEN.MEMBER.PARAM.CLASS	Reduced false positives
CS.NRE.FUNC.MIGHT	Reduced false positives
DBZ.GENERAL	New defects detected
FUNCRET.GEN	Reduced false positives
INVARIANT_CONDITION.UNREACH	Reduced false positives
LOCRET.ARG	Reduced false positives
LV_UNUSED.GEN	Reduced false positives
MISRA.ASSIGN.OVERLAP	New defects detected
MISRA.CAST.PTR.UNRELATED	Reduced false positives
MISRA.INIT.BRACES	Reduced false positives
MISRA.TYPEDEF.NOT_UNIQUE	Reduced false positives
RNPD. Deref	New defects detected

Checker	Description
UNREACH.GEN	Reduced false positives

### Enabled or disabled checkers

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

- JAVA.SV.EMAIL.HOST
- JAVA.SV.XML.INVALID
- PY3.E0001
- PY3.E0011
- PY3.E0013
- PY3.E0014
- PY3.E0015
- PY3.F0001
- PY3.F0002
- PY3.F0010
- PY3.F0011
- PY3.R0022
- PY3.W0012

## 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_c_all.tconf and cert_c_all_ja.tconf cert_c_rules.tconf and cert_c_rules_ja.tconf cert_cpp.tconf and cert_cpp_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"><li>▪ CERT MSC41-C</li></ul>
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	Added or modified checker mappings to the following weaknesses: <ul style="list-style-type: none"><li>▪ CWE-287</li></ul>
cwe_2021_top_25_cxx.tconf and cwe_2021_top_25_cxx_ja.tconf	Added or modified checker mappings to the following weaknesses: <ul style="list-style-type: none"><li>▪ CWE-276</li><li>▪ CWE-287</li></ul>
cwe_2021_top_25_java.tconf and cwe_2021_top_25_java_ja.tconf	Added or modified checker mappings to the following weaknesses:

Taxonomy	New/updated
	<ul style="list-style-type: none"> <li>▪ CWE-20</li> </ul>
cwe_2022_top_25_cxx.tconf and cwe_2022_top_25_cxx_ja.tconf	Added new taxonomies that map Klocwork checkers to the 2022 CWE Top 25 Most Dangerous Software Weaknesses.
cwe_all_cs.tconf and cwe_all_cs_ja.tconf	<p>Added or modified checker mappings to the following weaknesses:</p> <ul style="list-style-type: none"> <li>▪ CWE-562</li> <li>▪ CWE-672</li> <li>▪ CWE-896</li> </ul>
cwe_all_cxx.tconf and cwe_all_cxx_ja.tconf	<p>Added or modified checker mappings to the following weaknesses:</p> <ul style="list-style-type: none"> <li>▪ CWE-80</li> <li>▪ CWE-99</li> <li>▪ CWE-121</li> <li>▪ CWE-122</li> <li>▪ CWE-127</li> </ul>

Taxonomy	New/updated
	<ul style="list-style-type: none"> <li>▪ CWE-131</li> <li>▪ CWE-195</li> <li>▪ CWE-196</li> <li>▪ CWE-256</li> <li>▪ CWE-259</li> <li>▪ CWE-276</li> <li>▪ CWE-287</li> <li>▪ CWE-307</li> <li>▪ CWE-311</li> <li>▪ CWE-312</li> <li>▪ CWE-321</li> <li>▪ CWE-412</li> <li>▪ CWE-467</li> <li>▪ CWE-522</li> <li>▪ CWE-672</li> <li>▪ CWE-682</li> </ul>



Taxonomy	New/updated
	<ul style="list-style-type: none"> <li>▪ CWE-690</li> <li>▪ CWE-704</li> <li>▪ CWE-786</li> <li>▪ CWE-798</li> <li>▪ CWE-805</li> <li>▪ CWE-843</li> <li>▪ CWE-896</li> <li>▪ CWE-910</li> <li>▪ CWE-1335</li> </ul>
cwe_all_java.tconf and cwe_all_java_ja.tconf	<p>Added or modified checker mappings to the following weaknesses:</p> <ul style="list-style-type: none"> <li>▪ CWE-74</li> <li>▪ CWE-295</li> </ul>
disa_stig_v4_cxx.tconf and disa_stig_v4_cxx_ja.tconf disa_stig_v5_cxx.tconf and disa_stig_v5_cxx_ja.tconf	<p>Added or modified checker mappings to the following rules:</p>

Taxonomy	New/updated
	<ul style="list-style-type: none"> <li>▪ V-222432 (APSC-DV-000530)</li> <li>▪ V-222536 (APSC-DV-001680)</li> <li>▪ V-222551 (APSC-DV-001820)</li> <li>▪ V-222554 (APSC-DV-001850)</li> </ul>
disa_stig_v5_java.tconf and disa_stig_v5_java_ja.tconf	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ V-222555 (APSC-DV-001860)</li> </ul>
Helix QAC taxonomies	<p>The Helix QAC taxonomies have been updated to Helix QAC version 2023.2.</p>
jsf_av_rev_c_cpp.tconf and jsf_av_rev_c_cpp_ja.tconf	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ Rule 001</li> <li>▪ Rule 003</li> <li>▪ Rule 110</li> </ul>

Taxonomy	New/updated
<p>misra_c_2012_c90_all_checkers.tconf and misra_c_2012_c90_all_checkers_ja.tconf</p> <p>misra_c_2012_c90_certified.tconf and misra_c_2012_c90_certified_ja.tconf</p> <p>misra_c_2012_c99_all_checkers.tconf and misra_c_2012_c99_all_checkers_ja.tconf</p> <p>misra_c_2012_c99_certified.tconf and misra_c_2012_c99_certified_ja.tconf</p>	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ Rule 5.6</li> </ul>
<p>misra_c_2012_with_amd1_c90_all_checkers.tconf and misra_c_2012_with_amd1_c90_all_checkers_ja.tconf</p> <p>misra_c_2012_with_amd1_c90_certified and misra_c_2012_with_amd1_c90_certified_ja.tconf</p> <p>misra_c_2012_with_amd1_c99_all_checkers.tconf and misra_c_2012_with_amd1_c99_all_checkers_ja.tconf</p> <p>misra_c_2012_with_amd1_c99_certified.tconf and misra_c_2012_with_amd1_c99_certified_ja.tconf</p>	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ Rule 5.6</li> <li>▪ Rule 21.4</li> <li>▪ Rule 22.7</li> </ul>

Taxonomy	New/updated
<p>misra_c_2012_with_amd2_c11_all_checkers.tconf and misra_c_2012_with_amd2_c11_all_checkers_ja.tconf</p> <p>misra_c_2012_with_amd2_c11_certified.tconf and misra_c_2012_with_amd2_c11_certified_ja.tconf</p> <p>misra_c_2012_with_amd2_c90_all_checkers.tconf and misra_c_2012_with_amd2_c90_all_checkers_ja.tconf</p> <p>misra_c_2012_with_amd2_c90_certified and misra_c_2012_with_amd2_c90_certified_ja.tconf</p> <p>misra_c_2012_with_amd2_c99_all_checkers.tconf and misra_c_2012_with_amd2_c99_all_checkers_ja.tconf</p> <p>misra_c_2012_with_amd2_c99_certified.tconf and misra_c_2012_with_amd2_c99_certified_ja.tconf</p>	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ Rule 5.6</li> <li>▪ Rule 21.4</li> <li>▪ Rule 21.8</li> <li>▪ Rule 22.7</li> </ul>
<p>py.base.tconf and py.base_ja.tconf</p>	<p>Added or modified checker mappings to the following categories:</p> <ul style="list-style-type: none"> <li>▪ Basic</li> <li>▪ Classes</li> <li>▪ Exceptions</li> <li>▪ Imports</li> </ul>

Taxonomy	New/updated
	<ul style="list-style-type: none"> <li>▪ Standard Library Issues</li> <li>▪ Typecheck</li> </ul>

### Improvements to supported compilers

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

- Clang

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 2023.2. You need a new license to use the latest version of the product. Contact [license@perforce.com](mailto:license@perforce.com) to obtain a new license.

### 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

- Debian 11.7
- Red Hat Enterprise Linux 8.8
- Oracle Linux 8.7
- Amazon Linux 2 (2.0.20230515.0 Update)
- Ubuntu 22.04 to 22.04.2 LTS
- Fedora 38
- Eclipse 4.27 (2023-03)
- Android Studio Flamingo 2022.2.1 Patch 2
- Visual Studio 2017 version 15.9.54
- Visual Studio 2019 version 16.11.26
- Visual Studio 2022 version 17.6.1
- Visual Studio Code 1.78.2
- IntelliJ IDEA 2022.2.5, 2022.3.3
- CLion 2022.2.5, 2022.3.3, 2023.1 (up to 2023.1.3)
- Microsoft Edge 113.x
- Firefox 113.x
- Chrome 112.x
- Jenkins 2.406

In this release, we've ended support for

- Fedora 36
- Visual Studio Code 1.65.2 to 1.69.1
- Microsoft Edge 99.x to 102.x
- Firefox 98.x to 102.x

- Chrome 99.x to 102.x
- Safari 13.x

For the complete list of supported versions, see System Requirements.

### [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 2023.2

The following issues were fixed in Klocwork 2023.2.

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

### General issues

Number	Description
SUPPORT-23279	Improved the documentation for activating basic authentication.
SUPPORT-25376	Fixed an issue with the Visual Studio extension related to analyzing UTF-8 project source files.
SUPPORT-26782	Updated the documentation to clarify how to import a new taxonomy or file into a project.
SUPPORT-29262	Improved the documentation related to reporting a false positive.
00638250, SUPPORT-31370	Added documentation related to debugging a mail server configuration.
SUPPORT-36938	Updated the documentation to correct the location of the plugins directory for C# custom checkers.

Number	Description
00863136	Improved the documentation related to the detach process for kwinject.
SUPPORT-40791	Updated the documentation related to creating a C# build specification to add information related to using the devenv command line switch.
00624639, 00759634	Fixed an issue related to HIS Metrics configuration changes not being updated correctly in the portal.
SUPPORT-42230	Fixed an issue related to a duplicate defect reported in a C/C++ project.
00883983	Updated the documentation related to the kwprojcopy command.
00608653, SUPPORT-46087	Fixed an issue related to the Methods complexity report for C#.
SUPPORT-45301	Updated the documentation to clarify that wildcard asterisks function like Windows wildcards instead of regular expressions.
00604936	Improved the performance of the Visual Studio extension to reduce the initial load time for large projects, during which analysis could not be initiated.
00781469	Fixed an issue with the exact match operator in the issue list search not working properly with Kanji characters.

Number	Description
00747694	Updated the version of OpenJDK to 1.8.0.372.
00771681	Updated the documentation related to the suppression of macros.
00830180	Fixed an issue with the kwant command related to the location of the jvm.dll.
00831052	Added C# KAST syntax reference documentation.
00837012, 00867226	Updated the version of Apache Tomcat to 8.5.87.
00822505	Updated the documentation to clarify how Klocwork handles UTF-8 files that start with a BOM character.
00842526	Fixed an issue with the portal related to the list of projects loading slowly.
00844204, 00846346	Fixed an issue with the kwxsync and kwloadadb commands related to the consumption of memory.
00861419, 00895071	Fixed an issue with the calculation of the MAXLEVEL metric related to conditional statements.
00840198	Improved support for using the kwcheck command with the Visual Studio extension.

Number	Description
00877758	Removed an obsolete notation from a test case in the QDP.
00883613	Updated the checkers mapped to CWE-121 and CWE-122 in the CWE C/C++ "all" taxonomy.
00883626	Mapped the checker SV.TAINTED.XSS.REFLECTED to CWE-80 in the CWE C/C++ "all" taxonomy.
00884153	Mapped the checker CONC.DL to CWE-412 in the CWE C/C++ "all" taxonomy.
00886891	Mapped the checkers MISRA.CVALUE.IMPL.CAST, MISRA.CVALUE.IMPL.CAST.CPP, and MISRA.CONV.INT.SIGN to CWE-195 in the CWE C/C++ "all" taxonomy.
00886900	Mapped the checkers MISRA.CAST.INT.SIGN and MISRA.CONV.INT.SIGN to CWE-196 in the CWE C/C++ "all" taxonomy.
00886804	Mapped the checker ABV.GENERAL to CWE-127 in the CWE C/C++ "all" taxonomy.
00887063	Mapped the UFM.* checkers to CWE-672 in the CWE C/C++ "all" taxonomy.
00887077	Mapped the checkers NPD.FUNC.MUST, NPD.FUNC.MIGHT to CWE-690 in the CWE C/C++ "all" taxonomy.
00887619	Mapped the checker MISRA.CAST.OBJ_PTR_TO_INT.2012 to CWE-843 in the CWE C/C++ "all" taxonomy.

Number	Description
00887631	Mapped the checker SV.INCORRECT_RESOURCE_HANDLING.URH to CWE-910 in the CWE C/C++ "all" taxonomy.
00883389	Fixed an issue with the kwscansource command related to the packaging of a small number of .dll files in Windows.
00887069	Mapped the checker MISRA.SHIFT.RANGE to CWE-1335 in the CWE C/C++ "all" taxonomy.
00890202	Enhanced the C/C++ knowledge base related to the copy_from_user function for the checker NNTS.SRC.
00891548	Fixed an issue with CI build that included debug logs in the build directory.
00887613, 00887090	Updated the checkers mapped to CWE-321 and CWE-256 in the CWE C/C++ "all" taxonomy.
00891330	Fixed a Visual Studio 2022 issue related to NPD errors reported for unrelated source code.
00898344	Added a missing translated string related to the Kotlin checkers.
00898344	Updated the 2023.1 version of the What's new documentation to add that you can configure the Visual Studio extension to run with an external analysis engine such as the kwcheck command.
00915169	Removed an obsolete mention of Visual Studio 2012 from the clients.json file.

## Checker issues

Number	Description
00634766, SUPPORT-48842	Added support for MISRA 2012 AMD1: Rule 22.7 (Required): The macro EOF shall only be compared with the unmodified return value from any Standard Library function capable of returning EOF.
00767181, SUPPORT-31492	Reduced false positives for the checker MISRA.INIT.BRACES.
00639410	Added the community checker CXX.SIZEOF.CSTRING.
SUPPORT-44616	Reduced false positives for the checker ABV.STACK.
00608970, SUPPORT-44858	Improved defect detection for the checker ABV.GENERAL.
00708399	Improved defect detection for the checker MISRA.ASSIGN.OVERLAP.
00728415	Improved defect detection for the checker RNP.D. Deref.
00760082	Reduced false positives for the checker MISRA.TYPEDEF.NOT_UNIQUE related to the exception for MISRA C:2012 rule 5.6.

Number	Description
00787775	Reduced false positives for the checker MISRA.CAST.PTR.UNRELATED.
00800043, 00800044	Reduced false positives for the checker UNREACH.GEN.
00761003	Reduced false positives for the checker LOCRET.ARG.
00800045	Reduced false positives for the checker INVARIANT_CONDITION.UNREACH.
00887090	Added the C/C++ checker CXX.SV.PWD.PLAIN.
00834967	Reduced false positives for the checker CS.NRE.FUNC.MIGHT.
00886098	Reduced false positives for the checker LV_UNUSED.GEN.
00869691	Reduced false positives for the checker ANDROID.RLK.SQLOBJ.
00907179	Improved defect detection for the checker DBZ.GENERAL.

## 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

#### 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"><li>▪ <a href="#">Build specification templates</a></li></ul>

Feature	Details
<a href="#">Klocwork Static Code Analysis</a>	<ul style="list-style-type: none"> <li>"Show implementation", "Show declaration", and <a href="#">Source Cross-Reference</a></li> </ul>
<a href="#">Distributed analysis</a>	<ul style="list-style-type: none"> <li>Distributed analysis is not supported for C#.</li> </ul>

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

The following checkers listed in the Python taxonomy are not supported:

E0001	F0001	I0013	W0246
E0011	F0002	I0020	W3101
E0013	F0010	I0021	W4901
E0014	F0011	I0022	W4902

E0015	I0001	I0023	W4903
E0705	I0010	R0022	W4904
E1143	I0011	W0012	W4905

## 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).

### 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.

## Release Notes Klocwork 2023.1

These release notes cover Klocwork 2023.1 and include information about what's new in this release, and issues we've fixed since the last release,

### What's new in Klocwork 2023.1

Here are the highlights for Klocwork 2023.1. If you're upgrading, see the [Limitations](#) for items that affect how you use Klocwork.

#### Manage your Differential Analysis for CI/CD pipelines

Manage your differential analysis continuous integration builds by using the new CI Builds tab in Validate. By running CI builds, you can identify issues much faster and manage these issues in the same ways as you manage server issues, without having to run a full build. For more information see, [Managing your Differential Analysis for CI/CD pipelines](#). We've also added API actions to create, update, or delete CI builds, and to retrieve CI issue details. For more information, see [Issue and metric API examples](#).

#### C/C++

In this release we

- improved the speed of the analysis phase for projects by enhancing the parallelization of the analysis
- added support for several MISRA rules and increased coverage for MISRA C 2012 (up to AMD2) from 97.5% to 99%
- increased coverage for DISA STIG high severity rules

## C#

We improved support for version 8.0 of the C# language specification by adding support for

- const member declarations in interfaces
- readonly instance members
- static local functions
- default interface methods
- nullable reference types
- async streams
- using declarations
- disposable ref structs

## Java

We've added many new Java checkers that substantially improve our coverage of DISA STIG standards. See the Checkers section below for a complete list.

We enhanced support for Java 14 and have added partial support for Java 15:

- Improved support for Java 14 in PATH analysis includes improved support for switch expressions in PATH analysis.
- Klocwork now supports Java 15 for build integration commands such as kwant, kwgradle, and kwmaven and the Java 15 API for the Java knowledge base.

The kwandroid command now supports the --lang option that you can use generate separate build specification for C++ or Java.

### JavaScript, Kotlin, Python

We upgraded and improved JavaScript, Kotlin, and Python analysis engines and checkers, including adding syntax highlighting for code samples in the documentation to enhance usability.

The Python checkers taxonomy was modified so that you can provide checker arguments that customize Python checkers.

JavaScript analysis was also improved by adding support for the the .eslintignore file. This file lists ignored paths that should be skipped during JavaScript analysis.

### Visual Studio extension

You can now configure the Visual Studio extension to run with an external analysis engine such as the kwcheck command. If you have a large project, using the kwcheck command may reduce your analysis time. For more information, see [Analysis and Appearance tabs](#).

### Coding standards

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

- CWE
- DISA STIG
- Joint Strike Fighter Air Vehicle C++
- OWASP



- MISRA

### 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.SV.INSECURE_COOKIE	This C/C++ checker detects when an application uses cookies over a potentially unsecured network communication.
CXX.SV.PERSISTENT_COOKIE	This C/C++ checker detects when an application uses persistent cookies for tracking changes while on a website instead of using session cookies.
CXX.SV.XXE	This C/C++ checker detects when applications are vulnerable to XML-oriented attacks.
MISRA.INCL.LANG.FEATURES.2012	This MISRA checker provides support for MISRA C 2012 AMD2 Rules 1.4: Emergent language features shall not be used.

Checker	Description
MISRA.INCL.LANG.FEATURES.MT.2012	This MISRA checker provides support for MISRA C 2012 AMD2 Rules 1.4: Emergent language features shall not be used.
MISRA.LANG.FEATURES.2012	This MISRA checker provides support for MISRA C 2012 AMD2 Rules 1.4: Emergent language features shall not be used.
MISRA.LANG.FEATURES.MT.2012	This MISRA checker provides support for MISRA C 2012 AMD2 Rules 1.4: Emergent language features shall not be used.
MISRA.RESOURCES.FILE.OPEN_READ_WRITE.2012	This MISRA checkers provides support for MISRA C 2012 Rule 22.3: The same file shall not be open for read and write access at the same time on different streams.
MISRA.STDLIB.SYSTEM.2012_AMD2	This MISRA checker provides support for MISRA C 2012 Rule 21.21 : The Standard Library function <i>system</i> of <code>&lt;stdlib.h&gt;</code> shall not be used.
SV.CERT.INVALID	This Java checker detects when an X509 certificate is not validated and then generated by using Trust Anchors.

Checker	Description
SV.ECV.TRUSTMANAGER	This Java checker detects when an implementation of the X509TrustManager does not control the validity of the certificate, that is, no exception is raised.
SV.IL.SESSION	This Java checker detects when the session ID of the server or client is logged into application logs.
SV.PASSWD.HC.MINLEN	This Java checker detects when a hardcoded string is used by a method that accepts passwords or by a method that performs encryption.
SV.SESSION.FIXATION.COOKIE	This Java checker detects when a tainted value is used to set the JSESSIONID cookie.
SV.SPRING.FIXATION	This Java checker detects whenever session fixation protection is disabled.
SV.WEAK.KEYS.AES	This Java checker detects when the AES cryptographic algorithm is used with a key that is of insufficient size.
SV.WEAK.KEYS.DH	This Java checker detects when the DH cryptographic algorithm is used with a key that is of insufficient size.

Checker	Description
SV.WEAK.KEYS.DSA	This Java checker detects when the DSA cryptographic algorithm is used with a key that is of insufficient size.
SV.WEAK.KEYS.EC	This Java checker detects when the EC cryptographic algorithm is used with a key that is of insufficient size.
SV.WEAK.KEYS.RSA	This Java checker detects when the RSA cryptographic algorithm is used with a key that is of insufficient size.
SV.WEAK.TLS	This Java checker detects if a weak TLS protocol such as 1.0 or 1.1 is used.
SV.XSS.COOKIE.SECURE	This Java checker detects when a cookie that is used to store a session ID for a client's interaction with a website is not sent on a secure protocol such as HTTPS and SSL.

## Modified checkers

Checker	Description
ABV.GENERAL	New defects detected and reduced false positives
ABV.STACK	New defects detected
CL.SHALLOW.ASSIGN	New defects detected
CONC.DL	Overall improvements to the checker
DBZ family of checkers	New defects detected
FUNCRET.GEN	Reduced false positives
JD_LOCK	Overall improvements to the checker

### Enabled or disabled checkers

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

- SV.CERT.INVALID
- SV.ECV.TRUSTMANAGER
- SV.IL.SESSION
- SV.IL.SESSION.CLIENT
- SV.PASSWD.HC.MINLEN
- SV.SPRING.FIXATION

- SV.WEAK.KEYS.AES
- SV.WEAK.KEYS.DH
- SV.WEAK.KEYS.DSA
- SV.WEAK.KEYS.EC
- SV.WEAK.KEYS.RSA
- SV.WEAK.TLS
- SV.XSS.COOKIE.SECURE

### 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_c_all.tconf and cert_c_all_ja.tconf cert_c_rules.tconf and cert_c_rules_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ CERT EXP33-C</li> </ul>
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	Added or modified checker mappings to the following weaknesses: <ul style="list-style-type: none"> <li>▪ CWE-611</li> </ul>

Taxonomy	New/updated
cwe_2021_top_25_cxx.tconf and cwe_2021_top_25_cxx_ja.tconf	
cwe_all_cxx.tconf and cwe_all_cxx_ja.tconf	<p>Added or modified checker mappings to the following weaknesses:</p> <ul style="list-style-type: none"> <li>▪ CWE-539</li> <li>▪ CWE-611</li> <li>▪ CWE-614</li> </ul>
cwe_all_java.tconf and cwe_all_java_ja.tconf	<p>Added or modified checker mappings to the following weaknesses:</p> <ul style="list-style-type: none"> <li>▪ CWE-295</li> <li>▪ CWE-311</li> <li>▪ CWE-315</li> <li>▪ CWE-326</li> <li>▪ CWE-327</li> <li>▪ CWE-384</li> <li>▪ CWE-614</li> </ul>

Taxonomy	New/updated
	<ul style="list-style-type: none"> <li>▪ CWE-807</li> </ul>
<p>disa_stig_v4_cxx.tconf and disa_stig_v4_cxx_ja.tconf</p> <p>disa_stig_v5_cxx.tconf and disa_stig_v5_cxx_ja.tconf</p>	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ V-222577 (APSC-DV-002230)</li> <li>▪ V-222578 (APSC-DV-002240)</li> <li>▪ V-222596 (APSC-DV-002440)</li> <li>▪ V-222608 (APSC-DV-002550)</li> </ul>
<p>disa_stig_v4_java.tconf and disa_stig_v4_java_ja.tconf</p>	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ V-222396 (APSC-DV-000160)</li> <li>▪ V-222397 (APSC-DV-000170)</li> <li>▪ V-222536 (APSC-DV-001680)</li> <li>▪ V-222542 (APSC-DV-001740)</li> <li>▪ V-222543 (APSC-DV-001750)</li> <li>▪ V-222550 (APSC-DV-001810)</li> </ul>



Taxonomy	New/updated
	<ul style="list-style-type: none"> <li>▪ V-222555 (APSC-DV-001860)</li> <li>▪ V-222569 (APSC-DV-002010)</li> <li>▪ V-222571 (APSC-DV-002030)</li> <li>▪ V-222572 (APSC-DV-002040)</li> <li>▪ V-222585 (APSC-DV-002310)</li> <li>▪ V-222589 (APSC-DV-002350)</li> <li>▪ V-222596 (APSC-DV-002500)</li> <li>▪ V-222612 (APSC-DV-002590)</li> <li>▪ V-222641 (APSC-DV-003100)</li> </ul>
Helix QAC taxonomies	The Helix QAC taxonomies have been updated to Helix QAC version 2023.1.
jsf_av_rev_c_cpp.tconf and jsf_av_rev_c_cpp_ja.tconf	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ Rule 001</li> <li>▪ Rule 003</li> <li>▪ Rule 110</li> </ul>

Taxonomy	New/updated
<p>misra_c_2012_c90_all_checkers.tconf and misra_c_2012_c90_all_checkers_ja.tconf</p> <p>misra_c_2012_c90_certified.tconf and misra_c_2012_c90_certified_ja.tconf</p> <p>misra_c_2012_c99_all_checkers.tconf and misra_c_2012_c99_all_checkers_ja.tconf</p> <p>misra_c_2012_c99_certified.tconf and misra_c_2012_c99_certified_ja.tconf</p>	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ Rule 22.3</li> </ul>
<p>misra_c_2012_with_amd1_c90_all_checkers.tconf and misra_c_2012_with_amd1_c90_all_checkers_ja.tconf</p> <p>misra_c_2012_with_amd1_c90_certified and misra_c_2012_with_amd1_c90_certified_ja.tconf</p> <p>misra_c_2012_with_amd1_c99_all_checkers.tconf and misra_c_2012_with_amd1_c99_all_checkers_ja.tconf</p> <p>misra_c_2012_with_amd1_c99_certified.tconf and misra_c_2012_with_amd1_c99_certified_ja.tconf</p>	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ Rule 22.3</li> </ul>
<p>misra_c_2012_with_amd2_c11_all_checkers.tconf and misra_c_2012_with_amd2_c11_all_checkers_ja.tconf</p>	<p>Added or modified checker mappings to the following rules:</p>

Taxonomy	New/updated
<p>misra_c_2012_with_amd2_c11_certified.tconf and misra_c_2012_with_amd2_c11_certified_ja.tconf</p>	<ul style="list-style-type: none"> <li>▪ Rule 1.4</li> <li>▪ Rule 21.21</li> <li>▪ Rule 22.3</li> </ul>
<p>misra_c_2012_with_amd2_c90_all_checkers.tconf and misra_c_2012_with_amd2_c90_all_checkers_ja.tconf</p> <p>misra_c_2012_with_amd2_c90_certified and misra_c_2012_with_amd2_c90_certified_ja.tconf</p> <p>misra_c_2012_with_amd2_c99_all_checkers.tconf and misra_c_2012_with_amd2_c99_all_checkers_ja.tconf</p> <p>misra_c_2012_with_amd2_c99_certified.tconf and misra_c_2012_with_amd2_c99_certified_ja.tconf</p>	<p>Added or modified checker mappings to the following rules:</p> <ul style="list-style-type: none"> <li>▪ Rule 21.21</li> <li>▪ Rule 22.3</li> </ul>
<p>owasp_2021_10_cxx.tconf and owasp_2021_10_cxx_ja.tconf</p>	<p>Added or modified checker mappings to the following weaknesses:</p> <ul style="list-style-type: none"> <li>▪ A5</li> </ul>
<p>owasp_2017_10_java.tconf and owasp_2017_10_java_ja.tconf</p>	<p>Added or modified checker mappings to the following weaknesses:</p>

Taxonomy	New/updated
	<ul style="list-style-type: none"> <li>▪ A2</li> <li>▪ A3</li> <li>▪ A6</li> </ul>
owasp_2021_10_java.tconf and owasp_2021_10_java_ja.tconf	<p>Added or modified checker mappings to the following weaknesses:</p> <ul style="list-style-type: none"> <li>▪ A2</li> <li>▪ A4</li> <li>▪ A7</li> </ul>
py.base.tconf and py.base_ja.tconf	Renamed from python.py3.tconf and python.py3_ja.tconf.

### Improvements to supported compilers

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

- ARM Optimizing C/C++
- Clang
- IAR Systems C compiler/linker
- Lapis Technology CCU8

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

- Debian 11.6
- Oracle Linux 8.7
- Amazon Linux 2 (2.0.20230119.1 Update)
- Eclipse 2022-12 (4.26)
- Android Studio Electric Eel (2022.1.1 Patch 1)
- Visual Studio 2019 version 16.11.23
- Visual Studio 2022 version 17.4.4
- Visual Studio Code 1.65.2 (up to 1.75.1)
- IntelliJ IDEA 2022.3 (up to 2022.3.2)
- CLion 2022.3 (up to 2022.3.2)
- Microsoft Edge 99.x, to 110.x
- Firefox 98.x, to 110.x
- Chrome 99.x to 110.x
- Jenkins 2.391

In this release, we've ended support for

- FLEXlm/FlexNet Publisher
- macOS
- SUSE Enterprise 15
- Visual Studio Code 1.63.2 to 1.65.1
- IntelliJ IDEA 2016.x to 2018.x (up to 2018.3.6)
- Microsoft Edge 96.x to 98.x
- Firefox 96.x to 97.x
- Chrome 97.x to 98.x

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

## Fixed issues in Klocwork 2023.1

The following issues were fixed in Klocwork 2023.1.

### General issues

Number	Description
SUPPORT-35830	Enabled continuous analysis for C# on file save using the Visual Studio extension.
SUPPORT-41796	Added the definition of the BYTESSTACKSIZE metric to the documentation.
00637779, SUPPORT-44928	Improved the documentation related to the kwjsspec command.
00784304, 00826449, SUPPORT-47071	Fixed an issue with how permissions are handled for projects in streams so that permissions are assigned for specific streams and not on the parent project.
00801940, SUPPORT-47756	Improved support for the IAR Systems C compiler/linker.
00845964	Fixed an issue with the kwprojcopy command related to projects not appearing in the list of projects after being imported.

Number	Description
00686841	Fixed an issue with the Eclipse plug-in related to refactoring.
00717242	Reduced memory usage and improved the performance of the C/C++ analysis engine on a subset of projects.
00840198	Improved the performance of C/C++ analysis using the Visual Studio extension by providing a new analysis mode option.
00746102	Improved support for the Cadence Tensilica Clang-based compiler.
00842396, 00866296	Fixed an issue with the kwinject command related to the --variable option.
00811832	Improved documentation for the checkers MISRA.STDLIB.FENV.2012 and MISRA.STDLIB.FENV.MACRO.2012.
00798649, 00789970, 00810998, 00842058	Fixed a build issue related to a malformed object file.
00815996	Added the --lang option to the kwandroid command so that customers can specify either Java or C/C++.
00821320	Corrected the vulnerable code sample for the checker CS.NRE.FUNC.MIGHT.



Number	Description
00823479, 00829865	Fixed an issue with streams related to permissions not being retained properly after a server restart.
00634741	Improved support for the Lapis Technology CCU8 compiler.
00775508, 00775509	Reduced false positives for the checkers SV.CODE_INJECTION.SHELL_EXEC and SV.USAGERULES.PROCESS_VARIANTS.
00789772	Corrected a naming issue with how a Python checker was defined internally and improved the documentation related to customizing Python checkers.
00828822	Provided sample template files that customers can use to add custom links to the Validate portal.
00821192, 0821176, 00821186, 00821174	Added a small set of missing annotations to the QDP source code compliance package.
00830310, 00861374	Improved support for the ARM Optimizing C/C++ compiler.
00836636	Corrected the name of the --validate-server command-line option in the documentation for the desktop analysis plug-in package on Unix.
00829687, 00846221, 00848228, 00865149	Fixed a migration issue related to issue citations not migrating properly.

Number	Description
00859160	Corrected the syntax of the example for the kwadmin auto_delete_threshold option.
00834456	Updated documentation about how to use the kwwrap command with CMake.
00838622	Described the use of the environment variable "TMPDIR" in the Linux the installation notes.
00859080	Updated the documentation regarding role permissions.

### Checker issues

Number	Description
00683405, SUPPORT-30624, SUPPORT-33845	Reduced false positives for the checker ABV.GENERAL.
00845144, SUPPORT-42491	Reduced false positives for the checker FUNCRET.GEN.
00779810	Improved defect detection for the DBZ family of checkers.

Number	Description
SUPPORT-47013	Added support for MISRA C 2012 rules 1.4 and 21.21.
00603298	Improved defect detection for the checkers ABV.GENERAL and ABV.STACK.



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.